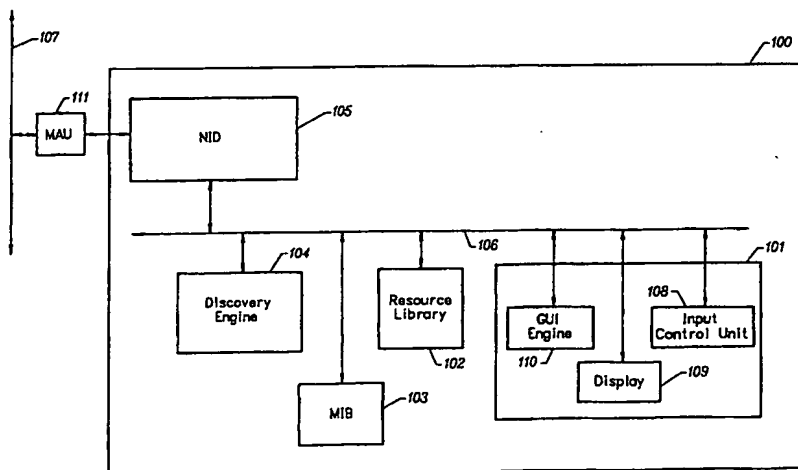




INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : G06F 13/00, H04L 12/56	A1	(11) International Publication Number: WO 99/61992 (43) International Publication Date: 2 December 1999 (02.12.99)
(21) International Application Number: PCT/US99/11614 (22) International Filing Date: 26 May 1999 (26.05.99) (30) Priority Data: 09/087,061 29 May 1998 (29.05.98) US (71) Applicant: THE BOEING COMPANY [US/US]; 2201 Seal Beach Boulevard, MCWSB-43, Seal Beach, CA 90740-1515 (US). (72) Inventors: SCHEUERMANN, Jim; 21485 Saratoga Hills Road, Saratoga, CA 95070 (US). HATLEY, Tom; 1116 Casaba Creek Court, San Jose, CA 95120 (US). TOY, Gene; 810 Wooded Hills Court, Fremont, CA 94539 (US). HANNER, Brian; 115 Del Prado Drive, Campbell, CA 95008 (US). MCGANN, L., Brian; 1812 Trout Gulch Road, Aptos, CA 95003 (US). JOHNSON, Rob; 8153 Quincy Street, Ventura, CA 93004 (US). MASTER, Paul; 796 Dartshire Way, Sunnyvale, CA 94087 (US). (74) Agent: MILLER, Mark, E.; Fliesler, Dubb, Meyer and Lovejoy LLP, Suite 400, Four Embarcadero Center, San Francisco, CA 94111-4156 (US).		(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG). Published <i>With international search report.</i> <i>Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>

(54) Title: SELF CONFIGURING NETWORK PERIPHERAL



(57) Abstract

A network peripheral (100) for interfacing to a communication network (107) includes a network interface device (105) for coupling the peripheral (100) to the network (107). The peripheral (100) configures the network interface device (105) to interpret information and determines whether the network interface device (105) is configured for interpreting information received from the network (107). Based on the determination, the network peripheral (100) modifies itself. When configured, the network interface device (105) includes a packet creation engine (122-2) for receiving network packets derived from the network (107) and creating cell packets (140) based on the network packets. The configured network interface device (105) also includes a set of nodes (180-1-5), which provide for configuring the network interface device (105). The processing engine (122) on the network interface device (105), such as the packet creation engine (122-2) and statistics engines (122-3), are formed in these nodes. In addition to processing engines (122), each node (200) includes a data transfer controller (202) and a task controller (203).

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon		Republic of Korea	PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakhstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

- 1 -

SELF CONFIGURING NETWORK PERIPHERAL

BACKGROUND OF THE INVENTIONA. Field of the Invention

The present invention is directed toward the field of network analysis and management.

5

B. Description of the Related Art

Dependence on communication networks is continually increasing. This increased dependence arises from a growing need for access to greater amounts of information and resources. In light of this increased dependence, there is a desire for communication networks to have high levels of performance and reliability. Network performance and reliability can be enhanced by analyzing and managing the network to ensure that its configuration and operation are optimally suited to the environment it is supporting.

10

However, the wide variety of existing and emerging technologies for use in communication networks makes the task of analyzing and managing a network very challenging. In fact, there is an increasingly wide variety of media, transport systems, and protocols that can be employed in communication networks. This makes it difficult for a network administrator to ensure that he or she has all the tools necessary for analyzing and managing a network.

15

20

A network administrator may be responsible for supporting an organization which employs several types of communication networks that are all linked together. For instance, each of the networks may be formed by a different physical medium, such as fiber optic cable or twisted pair cable, with

25

- 2 -

each network employing a different transport system, such as Synchronous Optical Network ("SONET"), Ethernet, or Token Ring.

5 Matters may be further complicated by the use of multiple protocols, within each transport system, for managing the information being transferred on the network. Such protocols include, but are not limited to, Asynchronous Transfer Mode ("ATM"), Frame Relay, Point-to-Point Over SONET, Packet Over SONET, and X.25. Within these protocols, several other layers of protocol can be nested for further organizing network information. Such additional protocols include TCP/IP and HTML.

10 A network analyzer is a tool that enables a network administrator to monitor and analyze the operation of a communication network. The network analyzer allows the network administrator to collect pertinent statistics relating to the information being passed on the network. Such statistics include, but are not limited to, the following: the number of errored packets, the types of
15 packet errors, and the magnitude of network traffic.

By gathering such statistics, a network administrator is able to identify potential trouble spots in the network. The administrator can then attempt to eliminate these trouble spots before the network's users encounter any difficulties with network performance or reliability.

20 Traditional network analyzers allow network administrators to add hardware modules for each type of communications network that is to be monitored by the network analyzer. For example, a SONET hardware module may be required for use with one network, while an Ethernet module is required for use with another network. As a result, a network administrator
25 needs to maintain many different types of hardware modules for supporting an organization with multiple network types. Some of the hardware modules may even have to be discarded, when new networking technologies are employed to replace existing networks.

- 3 -

However, each network analyzer hardware module is expensive. This makes it undesirable for a network administrator to purchase multiple network analyzer hardware modules, which may need to be discarded in the future. Accordingly, it would be desirable for a network administrator to be able to
5 purchase a single network analyzer hardware module that could be configured to support multiple types of existing networks, as well as networks that may be employed in the future.

The information that a network analyzer gathers about a communications network is stored in a management information base
10 ("MIB"), which includes fields for maintaining network statistics. In traditional network analyzers, the MIB includes fields for statistics relating to all the different types of communication networks that a network analyzer is anticipated to support.

A graphical user interface ("GUI") in the network analyzer retrieves
15 information from the MIB and displays it to a network administrator in a predetermined format. Traditionally, the GUI is unable to distinguish between MIB fields that correspond to the type of network that is presently being monitored by the analyzer and the remainder of the MIB fields. As a result, the network administrator interfaces with a GUI display that is
20 generalized for supporting a wide variety of networks, as opposed to a display that is tailored to the specific network that is presently being analyzed.

Accordingly, it would be desirable for a network analyzer to identify the type of network that is coupled to the analyzer and provide an indicator that enables a GUI to recognize the MIB fields that correspond to such a
25 network type. This would enable the GUI to provide a display that is optimized for the type of network being monitored. It would also be desirable for such a network analyzer to include a configurable hardware module,

- 4 -

which could be configured by the analyzer to support the identified type of network.

SUMMARY OF THE INVENTION

5 In accordance with the present invention, a network peripheral, such as a network analyzer, is provided for interfacing to a communication network. The network peripheral identifies the type of communication network to which it is coupled and configures itself for interfacing to such a network. The network peripheral also provides for maintaining a MIB that is tailored to the
10 communication network. Selected fields within the MIB are activated to enable a GUI to recognize and employ the MIB fields that correspond to the network.

The network peripheral includes a network interface device for coupling the peripheral to a communication network. In operation, the peripheral configures the network interface device to interpret information and determines
15 whether the network interface device is configured for interpreting information received from the communication network. Based on the determination, the network peripheral modifies itself. Such modifications include enabling fields within the MIB and providing information to MIB fields. Other modifications include configuring the network interface device to interpret information
20 according to a predetermined transport system or protocol.

When configured, the network interface device includes a packet creation engine for receiving network packets derived from the communication network. The packet creation engine transforms these network packets into cell packets, which contain data from the network packet and data derived from the network
25 packet. Additionally, the configured network interface device includes a set of statistics engines coupled to receive the cell packets. At least one of the statistics engines generates a set of statistics related to the cell packet.

- 5 -

The network interface device includes a set of nodes, which provide for configuring the network interface device. The processing engines on the network interface device, such as the packet creation engine and statistics engines, are formed in these nodes. In addition to processing engines, each node includes a data transfer controller and a task controller.

The data transfer controller controls the transfer of data between the node and a data storage unit. The task controller is coupled to the processing engines, the data transfer controller, and a system bus on the network interface device. The task controller controls communication between the system bus of the network interface device, the data transfer controller, and the processing engines.

BRIEF DESCRIPTION OF THE DRAWINGS

Further details of the present invention are explained with the help of the attached drawings in which:

Fig. 1 illustrates a network analyzer in accordance with the present invention.

Fig. 2 illustrates a block diagram of a network interface device in accordance with the present invention.

Fig. 3 illustrates a network interface device in one embodiment of the present invention.

Fig. 4 illustrates a network interface device in an alternate embodiment of the present invention.

Fig. 5 illustrates the format of a cell packet being generated on the network interface device shown in Fig. 4.

Fig. 6 illustrates a VCC control structure for the network interface device illustrated in Fig. 4.

Fig. 7 illustrates a generic flow control structure employed in conjunction with the cell statistics engine on the network interface device in Fig. 4.

- 6 -

Fig. 8 illustrates a VCC statistics structure employed in conjunction with the cell statistics engine on the network interface device in Fig. 4.

Fig. 9 illustrates the AAL statistics engine in Fig. 4 in one embodiment of the present invention.

5 Fig. 10 illustrates a network interface device containing a set of configurable nodes in one embodiment of the present invention.

Fig. 11 illustrates a network interface device containing a set of configurable nodes in an alternate embodiment of the present invention.

10 Fig. 12 illustrates a configurable node for use in a network interface device in accordance with the present invention.

Fig. 13 illustrates a sequence of operations performed by a discovery engine in a network peripheral in accordance with the present invention.

15 Fig. 14 illustrates a sequence of operations performed by a discovery engine to analyze a network peripheral in one embodiment of the present invention.

Fig. 15 illustrates a sequence of operations performed by a discovery engine to analyze a network peripheral in an alternate embodiment of the present invention.

20 Fig. 16 illustrates computer system hardware that may be employed to operate as a network peripheral in accordance with the present invention.

DETAILED DESCRIPTION

A. Network Peripheral

25 In accordance with the present invention, a network peripheral, such as a network analyzer, configures itself based on information that the peripheral receives from a communication network. In performing such a self configuration, the network peripheral identifies characteristics of incoming

- 7 -

information and adjusts itself accordingly. Such characteristics include, but are not limited to, transport systems and protocols associated with the information.

In configuring itself for interacting with the communication network, the network peripheral also establishes a customized MIB that is tailored to the communication network. The customized MIB provides a resource that is used by the peripheral's GUI to generate a display of network management information that is optimized for the communications network.

In order to provide for the self configuration capability, the network peripheral includes a configurable network interface device ("NID") and a discovery engine. The configurable NID provides for coupling the network peripheral to a communication network. The discovery engine provides for both the configuration of the NID and the customization of the MIB.

The operation of the NID and discovery engine eliminate the need for human interaction in the process of configuring the network peripheral to operate with a communication network. This reduces the need for the network administrator to have a complete working knowledge of the communications network, prior to actually analyzing the network.

Fig. 1 illustrates a network peripheral 100 in accordance with the present invention. The peripheral 100 includes a configurable network interface device 105, discovery engine 104, resource library 102, configurable management information base 103, and system interface 101, which are all coupled together within the peripheral 100 by a system bus 106. Although the system bus 106 is shown to be a single bus, the system bus 106 can actually be a set of buses that are coupled together to form a single logical bus.

The configurable NID 105 provides for coupling the network peripheral 100 to a communication network 107 and collecting statistics relating to the network 107. The NID is dynamically configurable, so that its configuration can be modified based on the type of communication network 107 to which it is

- 8 -

coupled. The NID's configurability is achieved through a set of configurable nodes that reside on the NID and can be programmed by the discovery engine 104.

5 In one embodiment of the present invention, each configurable node is formed by a field programmable gate array ("FPGA"), such as the XC4062XL. The XC4062XL is available from Xilinx Corporation and described in detail in XILINX The Programmable Logic Data Book 1998, which is also available from XILINX Corporation.

10 The configurable nodes are each configured to have a modular architecture, so that any node can be fully or partially modified without requiring the other nodes to be modified. This enables new functions to be programmed into a node, while the NID 105 is in operation. As a result, the NID 105 can be continually updated by the discovery engine 104 as new functionality is required by the network peripheral 100.

15 For instance, the NID 105 can be initially configured to support a SONET network employing an ATM protocol. During the operation of the network peripheral 100, the discovery engine 104 may recognize that the ATM protocol is not resident on the network 107. The discovery engine will then reconfigure a portion of one of the nodes to begin supporting a Frame Relay protocol, instead
20 of the ATM protocol. In a traditional network analyzer, such updating is not possible, since traditional analyzers do not provide for dynamic configuration.

As shown in Fig. 1, the NID 105 is coupled to a network 107 via an external medium attachment unit ("MAU") 111. The MAU 111 provides for converting signals from the network 107 into a data stream and clock signal that
25 are provided to the NID 105. Different media attachment units are employed for different network 107 media. Examples of network 107 media include, but are not limited to, OC-12 fiber optic cable, OC-3 fiber optic cable, coax, and twisted pair telephone wire. Although the NID 105 is shown to be coupled to the

- 9 -

network 107 via the external MAU 111, one with ordinary skill in the art will recognize that the MAU 111 can be integrated onto the NID 105, so that the NID 105 could be directly connected to the network.

5 In addition to configuring the NID 105, the discovery engine 104 collects network specific statistics from the NID 105. During operation, the discovery engine 104 retrieves resources from a resource library 102. The resource library 102 contains bit files for configuring the NID 105 and instructions to be executed by the discovery engine 104 in obtaining network related statistics from the NID 105.

10 The configurable MIB 103 maintains the network related statistics that are obtained by the discovery engine 104. The MIB 103 contains a predetermined set of fields for maintaining statistics that relate to a number of different types of communication networks. Fields within the MIB 103 are enabled by the discovery engine 104, as the discovery engine 104 determines what type of
15 communication network 107 is coupled to the NID 105. The discovery engine 104 then begins loading statistics into the enabled MIB fields, once the discovery engine 104 has retrieved these statistics from the NID 105.

In order to provide for displaying MIB information and receiving instructions from the peripheral's user, the peripheral 100 includes a system
20 interface 101. The system interface 101 includes an input control unit 108, display 109, and GUI engine 110, which are coupled to the system bus 106. The input control unit 108 enables the peripheral's user to communicate with the network peripheral 100, and the display 109 provides for a visual exchange of information between the peripheral's user and the peripheral 100.

25 The GUI engine 110 controls the transfer of data from the MIB 103 to the display 109. The GUI engine 110 can determine the type of network 107 that is coupled to the peripheral 100 by analyzing which MIB fields are enabled. The

- 10 -

GUI engine 110 is then able to optimize the display for the type of network 107 that is coupled to the peripheral 100.

B. Network Interface Device

5 Fig. 2 illustrates a block diagram of a configured NID 105 in accordance with the present invention. The NID 105 is configured to have a set of processing engines 122 that includes engines for performing operations utilizing information obtained from a predetermined type of communication network 107.

10 Such operations include, but are not limited to, the decoding of network information based on a transport system and set of protocols and collecting statistics that relate to the network information. In embodiments of the present invention, the set of processing engines 122 also provides for transmitting information onto the network 107. As will be described in greater detail below, the processing engines 122 are implemented on the NID 105 in a series of
15 configurable nodes that the discovery engine 104 configures.

The processing engines 122 are coupled to both memory 121 and a host interface 123. The memory 121 provides for the storage of various items, such as statistics that are collected by the processing engines 122 and control structures that provide for communication between the processing engines 122 and the
20 discovery engine 104. The host interface 123 provides for coupling the processing engines 122 to the system bus 106. In one embodiment of the present invention, the system bus 106 is comprised of a plurality of buses that are coupled together, with the bus directly connected to the NID 105 being a Peripheral Component Interconnect ("PCI") bus. The host interface 123 is also
25 coupled to the memory 121 to provide the discovery engine 104 with access to the memory 121.

The set of processing engines 122 also provides for coupling the NID 105 to a communication network 107. As shown in Fig. 2, the set of processing

- 11 -

engines 122 are coupled to the network 107 via a MAU 111. As described above, the MAU 111 can alternatively be integrated onto the NID 105.

Fig. 3 illustrates a configured NID 105 in one embodiment of the present invention. The set of processing engines 122 is configured to include a network
5 interface engine 122₁, packet creation engine 122₂, cell statistics engine 122₃, protocol 1 engine 122₄, and protocol 2 engine 122₅. Each of these engines 122₁₋₅ is coupled to the host interface 123 and a respective memory 121₁₋₅.

The network interface engine 122₁ provides for coupling the NID 105 to the communication network 107 via a MAU 111. Upon receiving information
10 from the network 107 through the MAU 111, the network interface engine 122₁ decodes the information based on a transport system. Examples of the transport system include, but are not limited to, the following: SONET, T-1, Ethernet, and Token Ring. In decoding the network information, the network interface engine 122₁ organizes the signals received from the network 107, via the MAU 111, into
15 a network packet that has a format corresponding to a specific transport system.

The network packet is provided to the packet creation engine 122₂, via a network packet bus 124, which couples the network interface engine 122₁ to the packet creation engine 122₂. The packet creation object 122₁ converts the network packet into a cell packet, which is then forwarded to other processing
20 engines 122 on the NID 105. The cell packet provides a means for the processing engines 122 to communicate with one another and have access to the data being retrieved from the network 107. As will be described in greater detail below, the cell packet includes information from the network and data derived from the network information.

25 The cell statistics engine 122₃ is coupled to the output of the packet creation engine 122₂ to receive the cell packet, via a cell packet bus 125 that couples the two engines 122₃, 122₂. The cell statistics engine 122₃ analyzes the cell packet to ascertain statistical information about the data received from the

- 12 -

network 107. These statistics are maintained by the cell statistics engine 122₃ in the memory 121₃ that is coupled to the cell statistics engine 122₃.

The cell statistics engine 122₃ provides the cell packet to the protocol 1 engine 122₄ on the cell packet bus 125 that couples the two engines 122₃, 122₄.
5 In embodiments of the present invention, the cell statistics engine 122₃ modifies the cell packet to reflect statistical information it has obtained, prior to providing the cell packet on the cell packet bus 125.

The protocol 1 statistics engine 122₄ generates statistical information related to the data in the cell packet, based on an assumption that the data in the
10 cell packet conforms to a predetermined protocol. The statistics are maintained by the protocol 1 engine 122₄ in the memory 121₄ coupled to the protocol 1 statistics engine 122₄.

The cell packet is then passed from the protocol 1 statistics engine 122₄ to a protocol 2 statistics engine 122₅, via the cell packet bus 125 coupling these
15 two engines 122₄, 122₅. In embodiments of the present invention, the protocol 1 engine 122₄ also modifies the cell packet in response to data within the cell packet, prior to providing the cell packet on the cell packet bus 125. The protocol 2 statistics engine 122₅ performs the same operations as the protocol 1 statistics engine 122₄ for a different predetermined protocol.

20 The statistics which are maintained by the various NID processing engines 121₁₋₅ in NID memory 121₁₋₅ are employed by the discovery engine 104 in configuring the NID 105 and maintaining the MIB 103. The discovery engine 104 accesses the NID memory 121₁₋₅ through the host interface 123 which is also coupled to each of the memories 121₁₋₅.

25 The discovery engine 104 determines whether the NID engines 122₁₋₅ are able to properly interpret the network data. If the network data is being properly interpreted, then the discovery engine 104 will retrieve and execute instructions from the resource library 102 that provide for gathering statistics from the NID

- 13 -

memory 121_{1,5} and updating the MIB 103. Otherwise, the engines which are improperly interpreting network data are replaced by the discovery engine 104. The processes performed by the discovery engine 104 for determining whether to modify the NID 105 processing engines 122 and for modifying these engines
5 122 will be discussed in greater detail below.

Fig. 4 illustrates a functional block diagram of the NID 105 in an alternate embodiment of the present invention. In this embodiment, a set of NID processing engines 122_{1,4} includes a standard chipset to operate as a network interface engine 122₁. As a result, the network interface engine 122₁ is not able
10 to be configured in the same manner as the other processing engines 122_{2,4}. As described above with reference to Fig. 3, processing engines 122_{2,4} are each coupled to a host interface 123 and a respective memory 121_{2,4}. Each memory 121_{2,4} is also coupled to the host interface 123.

In one embodiment of the present invention, the network interface engine
15 122₁ is a Saturn User Network Interface ("S/UNI") chipset, which is available from PMC-Sierra, Inc. under the part number PM5355. A detailed description of this chipset can be found in the Saturn User Network Interface Databook, which is available from PMC-Sierra, Inc. In such an embodiment, the remaining processing engines 122_{2,4} are implemented using a set of field programmable gate
20 arrays, such as Xilinx Corporation's XC4062XL.

When the network interface engine 122₁ is implemented using the S/UNI chipset, a SONET transport system is supported by the NID 105. A detailed description of the SONET transport system can be found in the following: 1) Bell Communications Research, *SONET Transport Systems Common Generic*
25 *Criteria GR-253-CORE*, Issue 1, December 1994; and 2) ANSI T1.105-1991 Telecommunications-Digital Hierarchy-Optical Interface Rates and Formats Specifications (SONET).

- 14 -

As shown in Fig. 4, the S/UNI network interface engine 122₁ is coupled to the communications network 107 via the MAU 111. In operation, the S/UNI network interface engine 122₁ receives communication network signals via the MAU 111 and formats the signals into a network packet in accordance with the SONET transport system. The network packet is provided by the S/UNI network interface engine 122₁ on a network packet bus 124, as described above with reference to Fig. 3.

The S/UNI network interface engine 122₁ also maintains a set of statistical information relating to the information it receives from the network 107. These statistics include indicators of signal activity on the network 107 and data frames that are lost due to a non-conformance with the SONET transport system. In addition to these statistics, the S/UNI chipset maintains a variety of other statistics, which are disclosed in greater detail in S/UNI Data Book.

The network packet bus 124 is coupled to the packet creation engine 122₂, so that the packet creation engine 122₂ can receive network packets provided by the S/UNI network interface engine 122₁. Upon receiving a network packet, the packet creation engine 122₂ generates a cell packet based on the network packet. The cell packet is created to include a header, which contains information about the cell packet, and a payload, which is the substantive information being transferred in the network packet.

In forming the cell packet, the packet creation engine 122₂ makes a determination of whether the information in the network data packet conforms to a predetermined networking protocol. Such protocols include, but are not limited to, ATM, Frame Relay, and TCP/IP. The result of such a protocol determination is maintained in the memory 121₂ for use by the discovery engine 104 in determining whether the packet creation engine 122₂ is properly configured.

- 15 -

In one embodiment of the present invention, the packet creation engine 122₂ is configured to support the ATM protocol. Fig. 5 illustrates the format of a cell packet 140 that is generated by the packet creation engine 122₂ in such an embodiment. The cell packet 140 includes a 16 byte header 150 and a 48 byte
5 payload 149. The payload 149 contains all of the payload data that is contained in a SONET network packet received on the network packet bus 124.

In the header 150, a Virtual Channel Connection ("VCC") field 141 contains a VCC stream number for the cell packet 140. The VCC stream number corresponds to the destination to which the cell packet 140 is directed.
10 In a SONET network packet, the destination is identified by a combination of a virtual path identifier ("VPI") and a virtual channel identifier ("VCI"). The VPI and VCI combine to form a 28 bit field that addresses a destination for the network packet.

The packet creation engine 122₂ performs a conversion operation to
15 convert the 28 bit VPI:VCI field into a 16 bit VCC stream number which is loaded into the VCC field 141. It is very unlikely that the entire possible number of 2^{28} destinations will appear on a single network. Therefore, the 2^{16} VCC stream numbers provide a sufficient number of possible identifiers for the different VPI:VCI combinations that are encountered.

20 The VCC field 141 is employed by other engines 122 on the NID 105. For example, other engines maintain statistics relating to each cell packet destination. Memory 121 on the NID 105 is set aside for these statistics for each possible destination. By reducing the number of possible destinations from 2^{28} to 2^{16} , the amount of memory and processing resources required for maintaining
25 these statistics is also reduced.

In performing the VPI:VCI to VCC conversion, in one embodiment of the present invention, the packet creation engine 122₂ performs a hashing operation. In the hashing operation, the VPI:VCI 28 bit address is split into a

- 16 -

19 bit address word and a 9 bit key word. In the memory 121₂, coupled to the packet creation engine 122₂, there are 2¹⁹ address reservoirs. Each address reservoir includes seven memory locations.

The 19 bit address word is used to address one of the VCC reservoirs.
5 The packet creation engine 122₂ then determines whether any of the memory locations in the addressed reservoir have already been assigned a VCC stream number. This determination is made by examining a set of flag bits that are set in the reservoir's memory locations.

If the flag bits are set to a 00 combination, then the memory location is
10 determined to be unassigned. If the flag bits are set to either 01 or 10, then the memory location is determined to include an assigned VCC stream number. If the memory location is determined to contain an assigned VCC stream number, then a key value is retrieved from the memory location. The key value is then compared to the key word that had been previously generated from the 28 bit
15 VPI:VCI address. If the key value from the memory location matches the key word, then a VCC stream number in the memory location has already been assigned to the VPI:VCI address. If desired, this value can then be retrieved from the memory location.

If none of the memory locations in the addressed reservoir are assigned
20 or none of the assigned memory locations have a corresponding key value, then a determination is made of whether any of the remaining memory locations in the reservoir are empty. If there are no remaining empty memory locations, then it will be impossible to obtain a VCC stream number for this particular VPI:VCI address. However, if there is an empty memory location, then the key word
25 derived from the VPI:VCI address is written into the empty memory location along with the next available VCC stream number. Since a VCC stream number is now assigned to this memory location, the memory location's flag bits are set

- 17 -

to 10 to indicate that the memory address has been assigned a VCC stream number.

5 In one embodiment of the present invention, bits 0-7 of the VPI are bits 11-18 of the reservoir address, and bits 0-10 of the VCI are bits 0-10 of the reservoir address. In such an embodiment, bits 8-11 of the VPI are bits 5-8 of the key word, and bits 11-15 of the VCI are bits 0-4 of the key word.

10 In an alternate embodiment of the present invention, the VPI:VCI address is a 24 bit quantity. In such an embodiment, bits 11-12 of the VCI and bits 0-10 of the VCI are bits 7-18 and 0-10, of the reservoir address, respectively. Bits 11-16 of the reservoir address are then constituted by bits 0-5 of the VPI. In such an embodiment, bits 5-8 of the key word are set to 0, while bits 13-15 of the VCI are bits 2-4 of the key word, and bits 6-7 of the VPI are bits 0-1 of the key word.

15 In addition to the VCC stream number, the cell packet header 150 also includes a stream disposition flag ("SDF") field 142. This field 142 contains indicators that are set by the packet creation engine 122, in accordance with directions it receives from the discovery engine 104. The SDF field 142 indicators are set to identify specific actions that are to be performed on the cell packet 140 by other processing engines and to identify certain characteristics of the cell packet 140. The SDF field 142 also includes indicators that are set by other NID processing engines 122 after such engines 122 perform operations employing the cell packet 140.

20 Table A below identifies SDF field 142 indicators.

- 18 -

TABLE A

Indicator	Description
IDL	This indicator signals that the cell packet is to be ignored by other processing engines 122 _{2,4} . Examples of when this indicator is set include: the VCC field 141 containing a predetermined value, an uncorrectable header error control error being detected, and the network packet being an idle cell.
BOF	The AAL processing engine 122 ₄ sets this indicator when the cell packet is a beginning cell of a frame.
EOF	The AAL processing engine 122 ₄ sets this indicator when the cell packet is an end cell of a frame.
CER	The AAL processing engine 122 ₄ sets this indicator on the last cell packet of a frame, if there has been a CSPDU error discovered.
SER	The AAL processing engine 122 ₄ sets this bit when the cell packet contains a SAR error.

5

- 19 -

In addition to the SDF field 142, the header 150 includes a secondary flag ("SF") field 143. The SF field 143 includes indicators that are set by the packet creation engine 122₂ to identify characteristics of the cell packet 140. Table B below identifies indicators in the SF field 143.

5

TABLE B

Indicator	Description
HECE	The packet creation engine 122 ₂ sets this indicator for a cell packet 140 when the network packet has an uncorrectable HEC error.
HECC	The packet creation engine 122 ₂ sets this indicator for a cell packet 140 when the network packet has a correctable HEC error.
AAL	The packet creation engine 122 ₂ sets this indicator to identify the AAL type for the cell packet 140.

10

An HEC error indicates that there has been a corruption of data received from the network 107 and placed in a network packet. In one embodiment of the present invention, the HEC error indicates that there is a corruption within the first five bytes of the network packet. The network interface engine 122₁, such as the S/UNI chipset, detects the presence of an HEC error when receiving data from the network 107. The network interface engine 122₁ determines whether a discovered HEC error is correctable or uncorrectable. The indication of the HEC error and whether it is correctable is then embedded into the network packet by the network interface engine 122₁ for detection by the packet creation engine 122₂. The packet creation engine 122₂ then sets the appropriate HEC indicator in the corresponding cell packet SF field 143 (Table B), if an HEC error is identified in the network packet.

15

20

- 20 -

The header 150 also includes a time stamp field 144, which indicates the time at which the cell packet 140 was received by the packet creation engine 122₂. Finally, the header 150 includes a network data packet ("NDP") field 145. The NDP field 145 contain the following information from the network packet from which the cell packet 140 is derived: generic flow control indicator, VPI indicator, VCI indicator, payload type indicator, and cell loss priority field. The above listed information from the network packet is described in detail in the ITU-T Recommendation I.361: Integrated Services Digital Network Overall Network Aspects and Function B-ISDN ATM Layer Specification, published by the International Telecommunications Union.

The data loaded into the cell packet header 150 by the packet creation engine 122₂ is provided in part by the discovery engine 104. The discovery engine 104 provides this information through a packet creation control status register and VCC control structure. The packet creation control status register is maintained on the NID 150 and includes the indicators set forth below in Table C.

TABLE C

Indicator	Description
GO	This indicator must be set for the operation of the packet creation engine to take place. When this indicator transitions from 0 to 1, processing will begin with the next complete network packet that is presented to the input of the packet generation engine 122 ₂ . When this indicator transitions from 1 to 0, processing of the current cell packet will be completed, and then processing will cease. In operation, the discovery engine sets and resets this indicator.

- 21 -

FRZ	This indicator is set to freeze the VCC learning operation, so that when a new VPI:VCI destination is encountered, no new VCC stream number is assigned.
AAL	This indicator must be set to enable the operation of the AAL processing engine 122 ₄ .
A/B	This indicator is used to switch all statistics processing between two buffer sets, known as buffer A and buffer B.
NVCC Streams	This indicator sets forth the number of active VCC stream numbers which have been encountered by the NID 105.

Fig. 6 illustrates the VCC control structure 160, which is maintained by the packet creation engine 122₂ in memory 121₂. For each VCC stream number that is supported by the NID 150, the VCC control structure 160 contains the following fields which correspond to the fields in a cell packet header 150: VCC field 161, SF field 162, SDF field 163, and NDP field 164. The packet creation engine 122₂ and discovery engine 104 communicate with one another by exchanging values in these fields.

The VCC field 161 contains a VCC stream number for a VPI:VCI combination that has been encountered by the NID 105 and assigned a VCC stream number by the packet creation engine 122₂. Upon assigning a VCC stream number to a VPI:VCI combination, the packet creation engine 122₂ loads the VCC stream number into a VCC field 161 in the VCC control structure 160 and loads the VPI:VCI combination into the NDP field 164 in the VCC control structure. This provides a cross reference of VCC stream numbers and VPI:VCI combinations that can be employed by the discovery engine 104 when formulating statistical data.

- 22 -

The SF field 162 contains an AAL indicator that is set by the discovery engine 104. When forming a cell packet 140 (Fig. 5), the packet creation engine 122₂ loads the cell packet's SF field 143 (Fig. 5) AAL indicator (Table B) with the AAL indicator value in the VCC control structure 160 entry having a VCC field 161 that matches the cell packet VCC field 141 (Fig. 5).

The VCC control structure 160 SDF field 163 contains an IDL indicator that is set by the discovery engine 104. When forming a cell packet 140 (Fig. 5), the packet creation engine 122₂ loads the cell packet's SDF field 142 (Fig. 5) IDL indicator (Table A) with the IDL indicator value in the VCC control structure 160 entry having a VCC field 161 that matches the cell packet VCC field 141.

One with ordinary skill in the art will recognize that additional indicators can be added to both the VCC control structure 160 and cell packet header 150 to enhance the functionality of the packet creation engine 122₂.

The cell statistics processing engine 122₃ receives cell packets from the packet creation engine 122₂ via the cell packet bus 125. Upon receiving a cell packet, the cell statistics processing engine 122₃ generates statistics related to the cell packet 140 and stores the statistics in memory 121₃. These statistics are available to the discovery engine 104 via the host interface 123, which is coupled to the memory 121₃.

The cell statistics engine 122₃ maintains a generic flow control ("GFC") count structure in memory 121₃ when it is configured for supporting the ATM protocol. The GFC count structure includes a pair of arrays. Fig. 7 illustrates the pair of arrays 166, 167 in the GFC count structure 165.

Each array 166, 167 includes 16 GFC counters, a correctable HEC error counter, and an uncorrectable HEC error counter. A GFC counter (GFC0-GFC15) is incremented when the cell statistics engine 122₃ receives a cell packet 140 (Fig. 5) that includes a corresponding GFC value in the NDP field 145.

- 23 -

The correctable HEC error count field is incremented when the cell statistics engine 122₃ receives a cell packet 140 (Fig. 5) with the correctable HEC error indicator (HECC) set in the SF field 143. The uncorrectable HEC error field is set when the cell statistics engine 122₃ encounters a cell packet 140 with the uncorrectable HEC error indicator (HECE) set in the secondary flag field 143.

There are two arrays 166, 167, so that the discovery engine 104 can retrieve statistics from one array, while the cell statistics engine 122₃ continues to maintain statistics counts for incoming cell packets. The control of which array is to be actively employed by the cell statistics engine 122₃ is controlled by the A/B indicator in the packet creation control status register, as shown in Table C above.

The cell statistics engine 122₃ also maintains a VCC statistics structure in the memory 121₃. Fig. 8 illustrates the VCC statistics structure 170. For each VCC stream encountered by the NID 105, a VCC field 171 and array 172 of statistics is maintained. Each VCC field 171 is loaded with a VCC stream number. Each array 172 includes two cell loss priority counters (CLP0, CLP1) and eight payload type counters (PTI0-PTI7).

The CLP0 counter for a VCC stream number is incremented by the cell statistics engine 122₃ upon receiving a cell packet 140 (Fig. 5) with a matching VCC field 141 and the cell loss priority indicator in the NDP field 145 not being set. The CLP1 counter for a VCC stream number is incremented by the cell statistics engine 122₃ upon receiving a cell packet 140 with a matching VCC field 141 and the cell loss priority indicator in the NDP field 145 being set.

Each payload type counter (PTI0-PTI7) corresponds to a different type of payload that can be employed in an ATM network packet. A payload type counter for a VCC stream number is incremented by the cell statistics engine 122₃ upon receiving a cell packet 140 (Fig. 5) that includes a corresponding VCC

- 24 -

stream number in the VCC field 141 and a corresponding payload type identified in the NDP field 145.

5 The AAL statistics engine 122₄ is responsible for maintaining statistics in memory 121₄ that relate to a set of ATM adaption layer ("AAL") protocols. In one embodiment of the present invention, the AAL statistics engine 122₄ supports AAL types 1, 3, 4, and 5. In such an embodiment, as shown in Fig. 9, the AAL statistics engine 122₄ really includes three separate processing engines 122₆₋₇ for supporting these AAL types. Specifications for AAL types 1, 3, 4 and 5 can be found in International Telecommunications Union Document I.363: B-ISDN
10 ATM Adaptation Layer (AAL) Specification, March 1993.

As shown in Fig. 9, the AAL statistic engine 122₄ includes an AAL1 statistics engine 122₆, an AAL34 statistics engine 122₇, and an AAL5 statistics engine 122₈. Each of these processing engines 122₆₋₈ is coupled to the host interface 123 and memory 121₄. The AAL1 statistics engine 122₆ receives cell
15 packets 140 from the cell statistics engine 122₃ via the cell packet bus 125, and provides cell packets 140 to the AAL34 statistics engine 122₇ via the cell packet bus 125. The AAL34 statistics engine 122₇ then provides the cell packets 140 to the AAL5 statistics engine 122₈ via the cell packet bus 125.

Upon receiving a cell packet 140 the AAL1 statistics engine 122₆,
20 determines whether it is supposed to process the cell packet 140. This determination is made by determining whether the AAL indicator in the cell packet's SF field 143 (Fig. 5, Table B) contains a predetermined value. In one embodiment of the present invention, the AAL1 statistics engine 122₆ will process the cell packet in response to multiple AAL indicator values.

25 If the cell packet 140 is not to be processed, the AAL1 statistics engine 122₆ immediately forwards the cell packet 140 to the outgoing cell packet bus 125, which provides the cell packet 140 to AAL34 statistics engine 122₇. When the cell packet 140 is processed by the AAL1 statistics engine 122₆, the AAL1

- 25 -

statistics engine 122₆ generates and records statistical information related to the cell packet 140 in an AAL1 statistics structure that is maintained in memory 122₄. The statistical information is generated under the assumption that the network packet from which the cell packet was derived is an AAL1 protocol type packet.

5

The AAL1 statistics structure includes an array of fields for each VCC stream number that is processed by the AAL1 statistics engine 122₆. The fields for each array are set forth below in Table D.

- 26 -

TABLE D

Field	Description
Correctable CRC Count	A count of all AAL1 cell packets having a corresponding VCC stream number and a correctable error in its CRC.
5 Errored Frame Count	A count of all AAL1 cell packets having a corresponding VCC stream number and an error.
SSE Error Count	A count of all AAL1 cell packets having a corresponding VCC stream number and a SAR sequence error.
10 SNP Error Counter	A count of all AAL1 cell packets having a corresponding VCC stream number and either a SAR, CRC error, or SAR parity error.
SRT Error Count	A count of all AAL1 cell packets having a corresponding VCC stream number and an SRTS error.

Each of these counters are incremented by the AAL1 statistics engine 122₆ upon processing a cell packet 140 that has a corresponding VCC stream number in the VCC field 141 (Fig. 5) and the condition being counted.

Upon receiving a cell packet 140, the AAL34 statistics engine 122₇ determines whether it is supposed to process the cell packet 140. This determination is made by determining whether the AAL indicator in the cell packet's SF field 143 (Fig. 5, Table B) contains a predetermined value. In one embodiment of the present invention, the AAL34 statistics engine 122₇ will process the cell packet in response to multiple AAL indicator values.

If the cell packet 140 is not to be processed, the AAL34 statistics engine 122₇ immediately forwards the cell packet 140 to the outgoing cell packet bus 125, which provides the cell packet 140 to AAL5 statistics engine 122₈. When the cell packet 140 is processed by the AAL34 statistics engine 122₇, the AAL34

- 27 -

statistics engine 122₇ generates and records statistical information related to the cell packet 140 in an AAL34 statistics structure that is maintained in memory 122₄. The statistical information is generated under the assumption that the network packet from which the cell packet 140 was derived is either an AAL3
5 or AAL4 protocol type packet.

The AAL34 statistics structure includes an array of fields for each VCC stream number that is processed by the AAL34 statistics engine 122₇. The fields for each array are set forth below in Table E.

- 28 -

TABLE E

Field	Description
Errors	This field indicates the detection of one or more of a predetermined set of errors in a cell packet having a corresponding VCC stream number. Such predetermined errors include the following: SAR Sequence Error; SAR CRC Error; CSPDU Length Error; CS Alignment Error; CS Segment Type Error
5 Frame Count	A count of all frames that are comprised of AAL34 cell packets having a corresponding VCC stream number.
Errored Frame Count	A count of all frames that are comprised of AAL34 cell packets having a corresponding VCC stream number and cause an error condition to be identified in the Error field.
Byte Count	A count of the total number of bytes in all AAL34 cell packets having a corresponding VCC stream number.
10 Error Count	A count of all AAL34 cell packets having a corresponding VCC stream number and an error.

Each of these fields are updated by the AAL34 statistics engine 122₇ upon processing a cell packet 140 that has a corresponding VCC stream number in the VCC field 141 (Fig. 5) and the condition being recorded.

Upon receiving a cell packet 140 the AAL5 statistics engine 122₈ determines whether it is supposed to process the cell packet 140. This determination is made by determining whether the AAL indicator in the cell packet's SF field 143 contains a predetermined value. In one embodiment of the present invention, the AAL5 statistics engine 122₈ will process the cell packet 140 in response to multiple AAL indicator values.

- 29 -

If the cell packet 140 is not to be processed, the AAL5 statistics engine 122_g does nothing with the cell packet 140. When the cell packet 140 is processed by the AAL5 statistics engine 122_g, the AAL5 statistics engine 122_g generates and records statistical information related to the cell packet 140 in an AAL5 statistics structure that is maintained in memory 122₄. The statistical information is generated under the assumption that the network data packet 140 from which the cell packet 140 was derived is an AAL5 protocol type packet.

The AAL5 statistics structure includes an array of fields for each VCC stream number that is processed by the AAL5 statistics engine 122_g. The fields for each array are set forth below in Table F.

- 30 -

TABLE F

Field	Description
Errors	This field indicates the detection of one or more of a predetermined set of errors in a cell packet having a corresponding VCC stream number. Such predetermined errors include CS CRC Errors.
5 Frame Count	A count of all frames that are comprised of AAL5 cell packets having a corresponding VCC stream number.
Errored Frame Count	A count of all frames that are comprised of AAL5 cell packets having a corresponding VCC stream number and an error.
10 Byte Count	A count of the total number of bytes in all AAL5 cell packets having a corresponding VCC stream number.

Each of these fields are updated by the AAL5 statistics engine 122₈ upon processing a cell packet that has a corresponding VCC stream number in the VCC field 141 (Fig. 5) and the condition being recorded.

15 In addition to maintaining the above described statistics structures, the engines 122₆₋₈ in the AAL statistics engine 122₄ also updates the SDF field 142 (Figs. 5, Table A) of a cell packet 140 that is being processed. As described above with reference to Table A, the AAL statistics engine 122₄ is responsible for updating the BOF, EOF, CER, and SER indicators for cell packets 140 that are processed. The updated SDF field 142 (Fig. 5) is provided in the cell packet 20 140 that is output onto the cell packet bus 125 by the engine 122₆₋₈ that is responsible for processing the cell packet 140.

- 31 -

As described above with respect to Figs. 3 and 4, the host interface 123 enables the discovery engine 104 to interface with the NID processing engines 122 and NID memory 121. For instance, the discovery engine gains access to the packet creation control status register (Table C) and VCC control structure 160 (Fig. 6) through the host interface 123 to control the operation of the NID processing engine 122. In embodiments of the present invention, the host interface 123 contains circuitry to provide an interface between the system bus 106 and processing engines 122 and memory 121.

In addition to the above-identified NID processing engines 122, new engines can be added to the NID 105 by the discovery engine 104. New engines can be added for collecting statistics that relate to other protocols such as TCP/IP and HTML. Additionally, engines can be added for performing functions such as capturing data in network data packets and issuing trigger signals upon the detection of predetermined events or packet characteristics.

Fig.10 illustrates a design for a configurable NID 105 for use in the network peripheral 100 shown in Fig. 1. The NID 105 shown in Fig. 10 can be configured by the discovery engine 104 to perform a number of different functional operations, including those described above with reference to Fig. 3 and Fig. 4.

The NID 105 includes a series of programmable nodes $180_{1,5}$ that are each coupled to a respective memory $121_{1,5}$ and the host interface 123. The host interface 123 is coupled to each of the memories $121_{1,5}$ through the configurable nodes $180_{1,5}$, once these nodes $180_{1,5}$ are configured. In one embodiment of the present invention, each configurable node $180_{1,5}$ is a field programmable gate array, such as the XC4062XL from Xilinx Corporation.

In one embodiment of the present invention, configurable node 180_1 will be configured by the discovery engine 104 to include a network interface engine 122_1 (Fig. 3). In order to support such an embodiment, configurable node 180_1

- 32 -

includes a set of inputs and outputs that are designated for configuration as a MAU interface bus 181. The MAU interface bus 181 will enable configurable node 180₁ to be coupled to a communication network 107 via a MAU 111. In one embodiment of the present invention, the MAU interface bus 181 includes a set of inputs for receiving data and a clock for synchronizing the input data and outputs for transmitting data and a synchronizing clock.

Configurable node 180₁ also includes a set of outputs that are designated for being configured as the above described network packet bus 124 (Figs. 3 and 4). Configurable node 180₂ has a series of inputs coupled to the network packet bus 124 outputs of configurable node 180₁ for receiving the network data packet bus 124. Configurable node 180₂ also includes a designated set of outputs for configuration as the cell packet bus 125 described above with reference to Figs. 3 and 4.

In one embodiment of the present invention, configurable node 180₂ is configured by the discovery engine 104 to include the packet creation engine 122₂, described above with reference to Figs. 3 and 4. Configurable nodes 180₃₋₅ each include a designated set of inputs for receiving cell packet bus 125 and a designated set of outputs for providing cell packet bus 125. Configurable nodes 180₃₋₅ are serially coupled together by the cell packet bus 125, so that the cell packet bus 125 outputs of configurable nodes 180₂, 180₃, and 180₄ are coupled to the cell packet bus 125 inputs of configurable nodes 180₃, 180₄, and 180₅, respectively.

In one embodiment of the present invention, the discovery engine 104 configures configurable nodes 180₃₋₅ to include the cell statistics engine 122₃ and AAL statistics engine 122₄, as described above with reference to Fig. 4. In an alternate embodiment of the present invention, configurable nodes 180₃₋₅ are each programmed to include different processing engines or sets of processing engines

- 33 -

to manage and analyze data that is received from the communication network 107.

Fig. 11 illustrates an alternate embodiment of the NID 105 in which the network interface engine 190 is implemented by the above-described S/UNI chip set. The S/UNI network interface 190 implements a MAU interface bus 181 for coupling the NID 105 to a communications network 107 via a MAU 111. The S/UNI network interface 190 also implements a network data packet bus 124 for providing data packets to configurable node 180₂ on the NID 105. Since the S/UNI network interface 190 is designed to support the SONET transport system, the network packets on the network packet bus 124 are formatted to conform with the SONET transport system.

In one embodiment of the present invention, the network packet bus 124 is able to both provide and receive network packets. Table G below identifies signals on a network packet bus 124 in such an embodiment.

- 34 -

TABLE G

Signal	Description
RDAT[15:0]	A data bus for providing network packets to configurable node 180 ₂ .
RXPRTY[1:0]	Parity bits for RDAT [15:0].
RCA	Signal that network packet is available from network interface engine.
RSOC	Signals the start of a network packet reception.
RFCLK	A clock signal synchronous with RDAT [15:0].
TDAT[15:0]	A data bus for receiving network packets from configurable node 180 ₂ .
TXPRTY[1:0]	Parity bits for TDAT [15:0].
TCA	Signals the transmission of a network packet to the network interface engine.
TFCLK	A clock signal synchronous with TDAT [15:0].
TSOC	Signals start of network packet being provided to the network interface engine.

The S/UNI network interface 190 also monitors data being received from the communication network 107 to determine characteristics of the data. For example, the S/UNI network interface 190 determines whether incoming data from the network 107 is being provided in accordance with SONET standards. Based on these determinations, the S/UNI network interface 190 generates and stores statistical information. This statistical information can be retrieved by the discovery engine 104 for use in updating the peripheral's MIB 103 (Fig. 1)

- 35 -

through the host interface 123, which is coupled to the S/UNI network interface 190.

Configurable nodes 180_{2-5} perform as described above with respect to Fig. 10. In order to interface with the S/UNI network interface 190, configurable node 180₂ has inputs coupled to at least the receive network packet bus 124 signals. This provides for the reception of network packets from the S/UNI network interface 190. Upon being configured by the discovery engine 104, these inputs in node 180₂ will operate in accordance with the receive data S/UNI chip set 190 network packet bus 124 signals.

10 Table H below identifies the signals that are employed on the cell packet bus 125, which is configured into each node 180_{2-5} in one embodiment of the present invention.

- 36 -

TABLE H

	Description
5	Input Signals
	IN_SYNC
	This signal is asserted to indicate the start of an incoming cell packet.
	IN_D[31:0]
	These signals form a data bus for receiving the data in the cell packet.
	IN_TAG[3:0]
	These signals provide supplemental information relating to the data on the IN_D[31:0] signals, as follows:
	0XXX Memory refresh indication asserted
	1XXX Memory refresh indication not asserted
	X111 No cell packet
	X110 Idle cell packet
	X101 Cell packet
	X100 Invalid cell packet
	Output Signals
10	OUT_SYNC
	This signal is asserted to indicate the start of a cell packet.
	OUT_D[31:0]
	These signals form a data bus for providing the data in the cell packet.
	OUT_TAG[3:0]
	These signals provide supplemental information

- 37 -

In embodiments of the present invention, as described above, each configurable node 180_{2,5} can be configured to contain one or more processing engines. When a configurable node contains multiple processing engines, each processing engine is linked by a bus, such as the above described cell packet bus 125.

In addition to processing engines 122 (Figs. 3-4), each configurable node 180_{1,5} includes an infrastructure for supporting the following interfaces: processing engines 122 and memory 121; processing engines 122 and host interface 123; and host interface 123 and memory 122. Fig. 12 illustrates a configurable node 200 that has been configured to include such an infrastructure, in accordance with the present invention.

After being configured, the configurable node 200 includes a task controller 203, a data transfer controller 202, and a set of processing engines 201_{1,3}. In operation, the configurable node is instructed to perform the operations of the processing engines 201_{1,3}, data transfer controller 202, and task controller 203 in response to instructions that are retrieved from a computer readable medium by the discovery engine 104 and provided to the configurable node 200.

The task controller 203 is coupled to each of the processing engines 201_{1,3} and the data transfer controller 202. The task controller 203 also includes a NID interface 204 for coupling the configurable node 200 to the host interface 123. In operation, the task controller 203 manages communications between the processing engines 201_{1,3}, data transfer controller 202, and host interface 123.

The data transfer controller 202 controls the transfer of data between the configurable node 200 and an external data storage unit, such as memories 121_{2,5} in Fig. 11. In order to interface to an external data storage unit, the data transfer controller 202 includes a memory interface 206. In one embodiment of the

- 38 -

present invention, the data transfer controller 202 and its memory interface 206 are designed to support a single external data storage unit. In an alternate embodiment of the present invention, the data transfer controller 202 and its memory interface 206 are designed to support multiple external data storage units.

5 Table I below illustrates a set of signals included in the memory interface 206, in one embodiment of the present invention, for supporting multiple external data storage units.

- 39 -

TABLE I

	Signal	Description
	M1_CS#	Select signal for a first data storage unit.
	M1_CS[3:0]#	Select signals for selecting a bank of the first data storage unit.
5	M1_CKE	Clock signal for a first data storage unit.
	M1_CKE[3:0]	Clock enable signals for enabling a clock for a bank of the first data storage unit.
	M1_CAS#	Column refresh signal for a first data storage unit.
	M1_RAS#	Row refresh signal for a first data storage unit.
	M1_WE#	Write enable for a first data storage unit.
10	M1_BA[1:0]	Memory bank address signals.
	M1_A[11:0]	Address signals for a first data storage unit.
	M1_MASK[3:0]	Signals for enabling the reading or writing of bytes in the M1_D[31:0] data signals.
	M1_D[31:0]	Data signals for a first data storage unit.
	M2_CS#	Select signal for a second data storage unit.
15	M2_CKE	Clock signal for a second data storage unit.
	M2_CAS#	Column refresh signal for a second data storage unit.
	M2_RAS#	Row refresh signal for a second data storage unit.
	M2_WE#	Write enable for a second data storage unit.
	M2_BA[1:0]	Memory bank address signals.
20	M2_A[11:0]	Address signals for a second data storage unit.
	M2_MASK[3:0]	Signals for enabling the reading or writing of bytes in the M2_D[31:0] data signals.
	M2_D[31:0]	Data signals for a second data storage unit.

- 40 -

In one embodiment of the present invention, the memory interface 206 is coupled to a set of synchronous dynamic random access memories provided by Samsung Electronics under the part number KM416S4030A. A detailed explanation of the operation of these memories can be found in the data sheet for the above-referenced part, which is also available from Samsung Electronics. In such an embodiment the memories are employed in a 4 bank by 1,048,576 word by 16 bit configuration.

The task controller 203 is coupled to the data transfer controller 202 by a data transfer interface 207. The data transfer interface 207 provides for passing data between the data transfer controller 202 and task controller 203. The interface 207 additionally provides for the transfer of addressing information for the data that is to be written to or read from the memory 121.

As described above, the processing engines 201_{1-3} perform a variety of functions, such as the functions described above for the processing engines 122 in Figs. 2-4. Each processing engine 122 is coupled to another processing engine via a bus for carrying a data stream, such as the network packet bus 124 and cell packet bus 125 that are described above. Each processing engine is also coupled to the task controller 203 via a respective engine interface 205_{1-3} . Each engine interface 205_{1-3} provides for the transfer of data between a processing engine 201_{1-3} and the task controller 203. Each engine interface 205_{1-3} also provides for transferring addressing information for referencing data storage locations in the memory 121, host interface 123 and processing engines 201_{1-3} . The task controller 202 is implemented to include at least as many engine interfaces 205 as are necessary for supporting the number of processing engines 201_{1-3} in the configurable node 200.

Through the task controller 203, the processing engines 201_{1-3} are able to interface with the data transfer controller 206 for storing data in memory. Such data includes the statistical information that is gathered by the processing engines

- 41 -

201_{1,3}. Examples of such statistical information include the information described above in Tables D-F when the processing engines 201_{1,3} are configured to operate as described above with respect to Fig. 4.

5 In order to further support the operation of the processing engines 201_{1,3}, the task controller also includes control registers (not shown) that enable the discovery engine 104 to communicate with the processing engines 201_{1,3}. One example of such a register is the packet creation control status register, which was described above with reference to Table C. The discovery engine 104 can issue instructions to the processing engines 201_{1,3} or receive information from the
10 processing engines 201_{1,3} through these registers. In an alternate embodiment of the present invention, registers for enabling communication between the processing origins 201_{1,3} and discovery engine 104 are included in the processing engine 201_{1,3}.

15 The NID's host interface 123 provides the vehicle through which the discovery engine 104 communicates with the task controller 203. Through the host interface 123, the discovery engine 104 is able to gain access to the configurable node 200 for accessing the control registers, processing engines 201_{1,3} and the memory 121 that is coupled to the node 200.

20 The task controller 203 enables the host interface 123 to access memory 121 by receiving instructions from the host interface 123 and exchanging data with the memory 121 through the data transfer controller 202 in response to such instructions. The task controller 203 enables communication between the host interface 123 and the processing engines 201_{1,3} by receiving instructions from the host interface 123 and performing appropriate operations with the processing
25 engines 201_{1,3} in response to such instructions.

The task controller 203 also enables the processing engines 201_{1,3} to exchange data with the memory 121 by receiving instructions from the processing engines 201_{1,3} and performing data exchange operations in response to such

- 42 -

instructions. In essence, the task controller 203 arbitrates between the processing engines 201_{1,3} and host interface 123 for access to resources that are either within or coupled to the configurable node 200.

5 The host interface 123 also provides an input window through which the configurable nodes can be configured by the discovery engine. The host interface receives bit streams from the discovery engine 104 that are to be downloaded into the configurable nodes 180_{1,5} (Figs. 10 and 11) for configuring the functionality of the nodes 180_{1,5}. The requirements for programming a field programmable gate array, which can be employed as a configurable node, are well known in the art and are described in XILINX The Programmable Logic Data Book 1998.

10 Although the configurable node 200 (Fig. 12) is shown to include three processing engines, one with ordinary skill in the art will recognize that the configurable node can include any number of processing engines.

15

C. Host System Operation

As described previously, the configurable NID 105 (Fig. 1) provides for coupling the network peripheral to a communication network 107. In addition to the NID 105, the network peripheral 100 also includes a discovery engine 104, system interface 101, MIB 103 and set of resource libraries 102.

20

The discovery engine 104 performs the dual function of configuring the NID 105 and maintaining the MIB 103. In operation, the discovery engine 104 provides the NID 105 with a configuration for a particular communication network 107. The discovery engine 104 then makes a determination of whether the configuration is proper and modifies the network peripheral 100 based on its determination. Such modifications include, but are not limited to, the following: reconfiguring the NID 105 upon learning that it is not properly configured; enabling a field or set of fields in the MIB 103 upon determining that the NID

25

- 43 -

105 is properly configured for a specific type of communications network; and gathering statistics from the NID 105 for processing and placement in enabled fields in the MIB 103.

Fig. 13 illustrates a sequence of operations performed by the discovery engine 104 in accordance with the present invention. Once the network peripheral 100 begins operating, the discovery engine 104 performs a default configuration of the NID 105 in step 210. In accordance with the present invention, such a configuration step 210 includes retrieving a predetermined set of bit files from the resource library 102 and employing those bit files to configure configurable nodes 180 (Figs. 10 and 11) on the NID 105. The configurable nodes are configured to include desired processing engines 122, as well as infrastructure functions that are required for supporting the operation of the processing engines 122.

In one embodiment of the present invention, the NID 105 includes the S/UNI chipset as the networking interface engine 122₁ (Fig. 4). In such an embodiment, the NID 105 is configured to include a packet creation engine 122₂, cell statistics engine 122₃, and AAL statistics engine 122₃, as described above with reference to Fig. 4. However, this is only one example of a possible default configuration, and one with ordinary skill in the art will recognize that a great number of default configurations are possible for supporting different transport systems and protocols.

Once the NID 105 receives a default configuration, the discovery engine 104 makes a determination of whether the NID 105 is actively receiving signals from a communication network 107 in step 211. In one embodiment of the present invention, this determination is made by querying the network interface engine 122₁ to determine whether it is receiving signals from a MAU 111.

When the S/UNI chip set is employed as a network interface engine 122₁, a loss of signal indicator, which is provided by the S/UNI chip set, is monitored

- 44 -

by the discovery engine 104. If the loss of signal indicator remains unasserted for a predetermined period of time, then a determination is made by the discovery engine 104 that the NID 105 is actively receiving signals from a network 107. Otherwise, a determination is made that the NID 105 is not
5 actively receiving network signals.

If a determination is made that network signals are not being received by the NID 105, then the signal detect step 211 is repeated by the discovery engine 104. Otherwise, the discovery engine 104 performs a network peripheral analysis operation in step 212. In the network peripheral analysis step 212, the
10 discovery engine 104 determines whether the NID 105 is configured to properly interpret information received from the communication network 107. Based on the determination, the discovery engine 104 will either modify the NID 105 or leave it in its present state, as part of the network peripheral analysis step 212.

Once the discovery engine 104 completes the network peripheral analysis
15 step 212, the discovery engine makes a determination of whether the network peripheral's MIB 103 needs to be updated in step 213. This determination is made based on the determination that was made in the network peripheral analysis step 212. For example, if it is determined that the NID 105 is configured properly in step 212, then the discovery engine 104 will decide to
20 update the MIB 103 by enabling MIB fields that correspond to the type of transport system and protocols that the NID 105 is configured to support.

If the discovery engine 104 makes the determination in step 213 that the MIB 103 is to be updated, then the discovery engine 104 proceeds to update the MIB in step 214. In updating the MIB 103, the discovery engine 104 can either
25 enable MIB fields or update already enabled MIB fields with new information. As described above, the enabling of MIB fields will be based on determinations that were made concerning the NID 105 in the peripheral analysis step 212. The

- 45 -

updating of MIB fields can also be performed using information from the peripheral analysis step 212, as will be described in greater detail below.

If the discovery engine, in step 213, determines that the MIB 103 does not need to be updated, then the discovery engine 104 makes a determination of whether it should modify its own functionality in step 215. This determination (step 215) is also made by the discovery engine 104 after the updating of the MIB 103 in step 214. In determining whether to modify its own functionality in step 215, the discovery engine 104 determines whether it is already performing all of the operations that are desirable. Such operations include the gathering of all information that is necessary to update all of the enabled fields in the MIB 103.

If the discovery engine 104 determines that it does not need to be modified in step 215, then the discovery engine 104 makes a determination of whether further peripheral analysis is desired in step 217. Otherwise, the discovery engine 104 modifies itself by retrieving sets of instructions that it wishes to execute from the resource library 102 in step 216. Once the discovery engine has updated itself by retrieving instructions from the resource library 102, it makes a determination of whether further peripheral analysis is desired in step 217. In one embodiment of the present invention, the instructions that are retrieved by the discovery engine 104 in step 216 are instructions that enable the discovery engine 104 to retrieve statistical information from the NID 105 for updating newly enabled MIB 103 fields.

If the discovery engine makes a determination in step 217 to continue peripheral analysis, the discovery engine 104 returns to the network peripheral analysis step 212 and repeats the steps following thereafter. Otherwise, the discovery engine 104 sequence of operations is done.

Fig. 14 illustrates a sequence of operations that are performed by the discovery engine 104 in the network peripheral analysis step 212 in one embodiment of the present invention. Upon beginning the network peripheral

- 46 -

analysis step 212, the discovery engine 104 queries the NID 105 in step 220. As a result of this query, the discovery engine retrieves statistical information that is stored in the NID 105 memories 121 by the processing engines 122. Examples of such statistical information are described above with reference to Figs. 3, 4, and 6-8 and Tables A, B and D-F. Much of this data is subsequently employed in updating the MIB 103 (step 214, Fig. 13).

Once the query 220 is completed, the discovery engine 104 makes a determination of whether the configurable nodes 180 on the NID 105 are configured to properly interpret information received from the communication network 107 in step 221. If it is determined that the configurable nodes 180 are properly configured for the network 107, then the network peripheral analysis step 212 is completed. Otherwise, the discovery engine 104 modifies the NID 105 configuration in step 222.

In one embodiment of the present invention, the NID 105 configuration is modified by reconfiguring the NID nodes 180 to include processing engines 122 for supporting a new transport system of protocol. The discovery engine 104 performs this modification by retrieving a new bit file from the resource library 102 and downloading it into one or more configurable nodes 180 to replace one or more of the existing processing engines 122. Once the NID is modified in step 222, the discovery engine has completed the peripheral analysis step 212.

Fig. 15 shows a more detailed sequence of operations that is performed by the discovery engine 104 when performing the network peripheral analysis step 212 in one embodiment of the present invention. As described with reference to Fig. 14, the discovery engine 104 first queries the NID 105 in step 220. Upon completing the query 220, the discovery engine 104 makes a transport system determination in step 230.

- 47 -

5 In making the transport system determination 230, the discovery engine 104 determines whether the NID 105 is configured for properly interpreting network data based on the transport system employed by the network 107. The transport system determination 230 is made by analyzing statistical data retrieved from the memory 121 in the query step 220. In one embodiment of the present invention, the statistical data is the statistics that are maintained by the network interface engine 122₁ (Figs. 3 and 4). Based on the statistics, the discovery engine 104 determines whether transport system errors are being detected in the reception of network data.

10 In one embodiment of the present invention, as described above with reference to Figs. 4 and 11, the network interface engine 122₁ is implemented by the S/UNI chip set. The S/UNI chip set maintains a statistic that indicates when an incoming packet from the network is not aligned with the proper format for the SONET transport system. The discovery engine 104 makes the transport system determination 230 by determining whether the S/UNI misalignment indicator remains unasserted for a predetermined period of time, such as 30 milliseconds. The predetermined time period can be implemented by ensuring that successive NID 105 query operations 220 are spaced apart by at least the predetermined period of time.

20 If the transport system determination indicates that the NID 105 does not support the transport system being used on the network 107 in step 230, then the NID 105 is modified in step 231. In the transport system modification step 231, the existing NID 105 network interface engine 122₁ is replaced with a network interface engine 122₁ for a new transport system.

25 In the transport system modification 231, the discovery engine 104 retrieves a bit file from the resource library 102 for a new network interface engine 122₁. The bit file is then loaded into a configurable node 180₁ (Fig. 10) by the discovery engine 104 to replace the existing network processing engine

- 48 -

122₁. Once the new network interface engine 122₂ is in place, the network peripheral analysis step 212 is done.

However, in some embodiments of the present invention the network interface engine 122₁ cannot be modified. One example is when the network interface engine 122₁ is implemented using the S/UNI chipset. In such an embodiment, the transport system modification step 231 is not executed. As a result (not shown), the peripheral analysis step 212 is completed, once the transport system determination 230 indicates that the network's transport system is not supported by the NID.

If the transport system determination 230 indicates NID 105 is supporting the network's transport system, then the discovery engine 104 makes a protocol determination in step 232. In making the protocol determination 232, the discovery engine 104 determines whether the NID 105 is able to properly interpret network data in accordance with a protocol nested within the transport system being supported by the NID 105. The protocol determination 232 is made by the discovery engine analyzing data that was retrieved in the NID query 220.

In one embodiment of the present invention, as described above with reference to Fig. 4, the NID 105 includes a packet creation engine 122₂ for receiving network packets and forming cell packets based on a predetermined protocol. The packet creation engine 122₁ also maintains statistics in a NID memory 121₂ to reflect whether received network data packets conform to the predetermined protocol. This statistical information is retrieved by the discovery engine 104 in the NID query 220. One example of such statistics is the HEC error indicators (Table B).

In performing the protocol determination 232, the discovery engine analyzes the retrieved packet creation engine 122₂ statistics. If the statistics indicate that network data packets are repeatedly not in conformance with the

- 49 -

predetermined protocol, then a conclusion is drawn that the packet creation engine 122₁ is not supporting the network's protocol.

5 If the protocol determination 232 indicates that the NID 105 does not support the protocol being used on the network 107 in step 232, then the NID 105 is modified in step 233. In the protocol modification step 233, the existing NID packet creation engine 122₂ is replaced with a packet creation engine 122₂ for a new protocol.

10 In the protocol modification 233, the discovery engine 104 retrieves a bit file from the resource library 102 for a new packet creation engine. The bit file is then loaded into a configurable node 180₂ (Figs. 10 and 11) by the discovery engine 104 to replace the existing packet creation engine 122₂. Once the new packet creation engine 122₁ is in place, the network peripheral analysis step 212 is done.

15 If it is determined in step 232 that the NID 105 supports the network's protocol, then the discovery engine 104 makes any necessary modifications to the VCC control structure (160, Fig. 6) that exists on the NID 105. As a first step, the discovery engine 104 determines whether there are any VCC control structure 160 entries that need to be modified in step 234. If modifications are to be made, then the discovery engine 104 selects one of the VCC control structure entries in step 236.

20 After selecting the VCC control structure entry, the discovery engine 104 makes a determination of whether a protocol type for the VCC control structure entry has been identified. If the protocol type has already been identified, then the discovery engine makes a determination, in step 240, of whether any other characteristics for the VCC control structure entry need to be identified.

25 If the protocol type has not been identified, then the discovery engine 104 identifies the protocol type in step 239. In an embodiment of the present invention, as described with respect to Fig. 4, an AAL type is ascertained by

- 50 -

evaluating values in AAL statistic structures (Tables D-F) that have been retrieved during the NID query in step 220. The discovery engine determines whether the VCC structure should contain an AAL1 type by calculating the ratio of the Errored Fame Count value in the AAL1 statistic structure (Table D) to the total number of AAL1 frames received by the NID 105 for the VCC stream number in the selected VCC control structure entry. The total number of AAL1 frames is obtained by adding the CLP0 and CLP1 fields in an entry in the VCC statistics structure 170 (Fig. 8) having a corresponding VCC stream number. The discovery engine 104 determines whether the ratio exceeds a predetermined threshold. If the threshold is not exceeded, then the discovery engine identifies the AAL1 type as the correct protocol type.

If the AAL1 type is not identified as correct, then the discovery engine determines whether the VCC structure should contain an AAL34 type. The discovery engine 104 calculates the ratio of the Errored Fame Count value to the Frame Count value in the AAL34 statistic structure (Table E) and determines whether this ratio exceed a predetermined threshold. If the threshold is not exceeded, then the discovery engine identifies the AAL34 type as the correct protocol type.

If the AAL34 type is not identified as correct, then the discovery engine determines whether the VCC structure should contain an AAL5 type. The discovery engine 104 calculates the ratio of the Errored Fame Count value to the Frame Count value in the AAL5 statistic structure (Table F) and determines whether this ratio exceed a predetermined threshold. If the threshold is not exceeded, then the discovery engine identifies the AAL5 type as the correct protocol type.

Once the protocol type has been identified, in step 239, the discovery engine makes a determination of whether other characteristics for the VCC control structure entry need to be identified. If other characteristics are to be

- 51 -

identified, then these identifications are made in step 241. Otherwise, the discovery engine 104 determines whether there are any more VCC control structures which still need modification in step 234. The determination in step 234 is also repeated after additional characteristics are identified in step 241.

5 Once it is determined that there are no VCC structures that need further modification in step 234, the discovery engine 104 makes a determination of whether any updates are to be made to the VCC control structure in step 235. Such updates are to be made when new protocols or characteristics are identified in steps 239 and 241, respectively. If no VCC control structure updates are to
10 be made, then the network peripheral analysis step 212 is done.

 Otherwise, the discovery engine 104 proceeds to update VCC control structures in step 237. In updating the VCC control structure 160 (Fig. 6), the discovery engine 104 writes each newly discovered value from steps 239 and 241 into the corresponding field in a VCC control structure entry on the NID 105.
15 For example, when a new AAL type is determined for a VCC control structure entry, the discovery engine 104 writes the AAL type into the SF field 162 (Fig. 6) in the VCC control structure 160 entry having the corresponding VCC stream number. Once the VCC control structure updating 237 is completed, the network peripheral analysis step 212 is completed.

20 When protocol types or other characteristics are identified (239, 241), such identifications operate as indicators that the MIB 103 and/or discovery engine 104 should be modified. For example, the identification of a protocol type signals that corresponding statistical fields in the MIB 103 should be enabled and the discovery engine 104 should be modified to update these fields
25 continually.

 One with ordinary skill in the art will recognize that additional protocol determination steps and protocol modification steps can be added to the above

- 52 -

sequence of operations. This may be done to support networks that have multiple protocols nested within the transport system or within other protocols.

As the discovery engine 104 continues to modify the NID 105 and update the network peripheral 100, the system interface 101 continually provides
5 updated network related information to the peripheral's user. In providing the peripheral's user with such information, the system interface 101 retrieves information that the discovery engine 104 has made available in the MIB 103 and processes this information with the graphical user interface 110. The graphical user interface 110 employs the MIB 103 information to generate displays of
10 network related information that are optimized for the type of transport system and protocols that are resident on the network 107.

D. Network Peripheral Hardware

Fig. 16 illustrates a high level block diagram of a general purpose
15 computer system 300, which is employed in one embodiment of the present invention to form a network peripheral 100. Accordingly, the computer system 300 can be employed for performing a number of processes, including those illustrated in Figs. 13-15.

The computer system 300 contains a processing unit 301, main memory
20 302, and an interconnect bus 310. The processing unit 301 can contain either a single microprocessor or a plurality of microprocessors for configuring the computer system 300 as a multi-processor system. The processing unit 301 is employed to operate as the discovery engine 104, by retrieving and executing instructions from a computer readable medium, such as a main memory 302,
25 mass storage device 303, or portable storage medium drive 306.

The main memory 302 stores, in part, instructions and data for execution by the processing unit 301. Such data includes the MIB 103 and resource library 102. If a process, such as the processes illustrated in Figs. 13-15, is wholly or

- 53 -

partially implemented in software, the main memory 302, in one embodiment of the present invention, stores the executable instructions for implementing the process when the computer is in operation. The main memory 302 can include banks of dynamic random access memory (DRAM) as well as high speed cache
5 memory.

The computer system 300, in embodiments of the present invention, further includes a mass storage device 303, peripheral device(s) 304, portable storage medium drive(s) 306, input control device(s) 305, a graphics subsystem 307, and an output display 308. For purposes of simplicity, all components in the
10 computer system 300 are shown in Figure 16 as being connected via the bus 310. However, the computer system 300 can be connected through one or more data transport means. For example, the processor unit 301 and the main memory 302 can be connected via a local microprocessor bus, and the mass storage device 303, peripheral device(s) 304, portable storage medium drive(s) 306, and
15 graphics subsystem 307 can be connected via one or more input/output (I/O) busses.

The mass storage device 303, which can be implemented with a magnetic disk drive or an optical disk drive, is a non-volatile storage device for storing data and instructions for use by the processor unit 301. In software embodiments
20 of the present invention, the mass storage device 303 stores the instructions executed by the computer system 300 to perform processes for the discovery engine 104. The mass storage device 303 can also act as a storage medium for the MIB 103 and resource library 102.

The portable storage medium drive 306 operates in conjunction with a
25 portable non-volatile storage medium, such as a floppy disk, a compact disc read only memory (CD-ROM), or an integrated circuit non-volatile memory adapter (i.e. PC-MCIA adapter) to input and output data and code to and from the computer system 300. In one embodiment, the instructions for enabling the

- 54 -

computer system to execute processes, such as those illustrated in Figs. 13-15, are stored on such a portable medium, and are input to the computer system 300 via the portable storage medium drive 306.

5 The peripheral device(s) 304 can include any type of computer support device, such as an input/output (I/O) interface, to add additional functionality to the computer system 300. For example, the peripheral device(s) 304 can include a communications controller, such as a network interface card or integrated circuit, for interfacing the computer system 300 to a communications network. Instructions for enabling the computer system 300 to perform processes, such as
10 those illustrated in Figs. 13-15, can be downloaded into the computer system's main memory 302 over a communications network. In one embodiment of the present invention, the configurable network interface device 105 is a peripheral device 304.

15 The input control device(s) 305 provide the input control unit 108 for the network peripheral 100. The input control device(s) 305 can include an alphanumeric keypad for inputting alphanumeric and other key information, a cursor control device, such as a mouse, a trackball, stylus, or cursor direction keys.

20 In order to display textual and graphical information, such as the GUI engine's representation of the MIB 103 information, the computer system 300 contains the graphics subsystem 307 and the output display 308. The output display 308 can include a cathode ray tube (CRT) display or liquid crystal display (LCD). The graphics subsystem 307 receives textual and graphical information, and processes the information for output to the output display 308. The graphics
25 subsystem 307 and output display 308 combine to form the display 109 for the network peripheral 100.

The process steps and other functions described above with respect to embodiments of the present invention are implemented as software instructions,

- 55 -

in one embodiment of the present invention. More particularly, the configurable node bit files, the process steps illustrated in Figs. 13-15, as well as the operations performed by the discovery engine 104 and GUI engine 110, are implemented as software instructions. For the preferred software implementation, the software includes a plurality of computer executable instructions for implementation on a general purpose computer system. Prior to loading into a general purpose computer system, the software instructions may reside as encoded information on a computer readable medium, such as a magnetic floppy disk, magnetic tape, and compact disc read only memory (CD - ROM). In one hardware implementation, circuits may be developed to perform the process steps and other functions described herein.

Although the invention has been described above with particularity, this was merely to teach one of ordinary skill in the art how to make and use the invention. Many modifications will fall within the scope of the invention, as that scope is defined by the following claims.

- 56 -

CLAIMS

What is claimed is:

1. A computer implemented method for configuring a network peripheral, wherein the network peripheral includes a network interface device for coupling the network peripheral to a communication network, said method comprising the steps of:
 - 5 (a) configuring the network interface device for interpreting information;
 - (b) determining whether the network interface device is configured for interpreting information received from the communication network; and
 - (c) modifying the network peripheral based on the determination in
10 said step (b).
2. The method of claim 1, wherein said step (c) includes the step of:
providing the network interface device with a new configuration for interpreting information.
3. The method of claim 2, wherein said step of providing includes the step of:
configuring the network interface device to interpret information from the communication network in accordance with a predetermined transport system.
4. The method of claim 3, wherein the predetermined transport system is a transport system selected from a group of transport systems consisting of asynchronous transfer mode, Point-to-Point Over SONET, and Packet Over SONET.

- 57 -

5. The method of claim 2, wherein said step of providing includes the step of:

configuring the network interface device to interpret information from the communication network in accordance with a predetermined protocol.

6. The method of claim 3, wherein the predetermined protocol is a protocol selected from a group of protocols consisting of ATM, frame relay, TCP/IP and HTML.

7. The method of claim 1, wherein the network peripheral includes a management information base having a plurality of fields for maintaining data related to the communication network, wherein said step (c) includes the step of:

5 modifying the management information base based on the determination in said step (b).

8. The method of claim 7, wherein said step of modifying the management information base includes the step of:

enabling a field in the management information base for maintaining data related to the communication network.

9. The method of claim 7, wherein said step of modifying the management information base includes the step of:

entering data into a field in the management information base.

10. The method of claim 1, wherein the network peripheral includes a discovery engine for performing step (a), step (b), and step (c), and wherein said step (c) includes the step of:

- 58 -

5 (b). modifying the discovery engine based on the determination in said step

11. The method of claim 1, wherein said step (b) includes the steps of:
 querying the network interface device to obtain data; and
 deciding whether the network interface device is configured for
 interpreting information received from the communication network, based on the
5 data obtained from said step of querying the network interface device.

12. The method of claim 11, wherein said step of deciding includes
 the step of:
 determining whether the network interface device is configured for
 interpreting information from the communication network in accordance with a
5 transport system employed on the communication network.

13. The method of claim 12, wherein said step of determining includes
 the step of:
 determining whether a framing error is detected by the network interface
 device within a predetermined period of time.

14. The method of claim 12, wherein said step (c) includes the step
 of:
 configuring the network interface device to interpret information from the
 communication network in accordance with a predetermined transport system,
5 if said step of determining indicated that the network interface device is not
 configured for interpreting information from the communication network in
 accordance with a transport system employed on the communication network.

- 59 -

15. The method of claim 11, wherein said step of deciding includes the step of:

determining whether the network interface device is configured for interpreting information from the communication network in accordance with a protocol employed on the communication network.

16. The method of claim 15 wherein said step of determining includes the step of:

determining whether a cell delineation error is detected by the network interface device within a predetermined period of time.

17. The method of claim 15, wherein said step (c) includes the step of:

configuring the network interface device to interpret information from the communication network in accordance with a predetermined protocol, if said step of determining indicated that the network interface device is not configured for interpreting information from the communication network in accordance with a protocol employed on the communication network.

18. The method of claim 1, further including the step of:

(d) identifying a characteristic for a destination that is identified in a packet that is received by the network interface device from the communication network.

19. The method of claim 18, wherein said step (d) include the steps of:

querying the network interface device to obtain data; and

- 60 -

5 ascertaining the characteristic based on the data obtained from said step
of querying the network interface device.

20. The method of claim 19, wherein the characteristic is a protocol
type.

21. The method of claim 20, wherein said step of ascertaining includes
the step of:

determining whether a ratio of errored frames for the destination to total
frames for the destination exceeds a predetermined threshold.

22. The method of claim 20 further including the step of:

(e) informing the network interface device of the protocol type that
is ascertained.

23. A computer readable medium having a set of instructions stored
therein for enabling a computer to configure a network peripheral, wherein the
network peripheral includes a configurable network interface device for coupling
the network peripheral to a communication network, said set of instructions
5 including:

a first set of instruction, which when executed by the computer cause the
computer to configure the network interface device for interpreting information;

a second set of instruction, which when executed by the computer cause
the computer to determine whether the network interface device is configured for
10 interpreting information received from the communication network; and

a third set of instruction, which when executed by the computer cause the
computer to modifying the network peripheral based on the determination made
in response to said second set of instructions.

- 61 -

24. The computer readable medium of claim 23, wherein said third set of instructions includes:

5 a fourth set of instructions, which when executed by the computer cause the computer to provide the network interface device with a new configuration for interpreting information.

25. The computer readable medium of claim 23, wherein said third set of instructions includes:

5 a fourth set of instructions, which when executed by the computer cause the computer to modify the management information base, based on the determination made in response to said second set of instructions.

26. The computer readable medium of claim 23, wherein said second set of instructions includes:

5 a sixth set of instructions, which when executed by the computer cause the computer to query the network interface device to obtain data; and
a seventh set of instructions, which when executed by the computer cause the computer to decide whether the network interface device is configured for interpreting information received from the communication network, based on data obtained in response to said sixth set of instructions.

27. The computer readable medium of claim 26, wherein said third set of instructions includes:

5 an eighth set of instructions, which when executed by the computer cause the computer to configure the network interface device to interpret information from the communication network in accordance with a predetermined protocol, in response to the decision made in response to said seventh set of instructions.

- 62 -

28. The computer readable medium of claim 23, wherein said set of instructions further includes:

5 a ninth set of instructions, which when executed by the computer cause the computer to identify a characteristic for a destination that is identified in a packet received by the network interface device from the communications network; and

a tenth set of instructions, which when executed by the computer cause the computer to inform the network interface device of the characteristic that is identified in response to said ninth set of instructions.

29. A configurable network interface device for coupling a network peripheral to a communication network, said network interface device comprising:

5 a packet creation engine for receiving a set of information derived from the communication network and transforming the set of information into a cell packet, wherein the cell packet contains the set of information and data derived from the set of information; and

10 a set of statistics engines coupled to receive the cell packet, wherein at least one statistics engine in the set of statistics engines generates a set of statistics related to the cell packet.

30. The configurable network interface device of claim 29, further including:

a control structure containing a set of entries, wherein each entry includes a cell packet identifier and a set of related data.

- 63 -

31. The configurable network interface device of claim 30, wherein the packet creation engine is adapted for loading the cell packet with the cell packet destination identification and a portion of the set of related data from an entry in the control structure that corresponds to the set of information.

32. The configurable network interface device of claim 29, wherein the packet creation engine is adapted for loading a time stamp into the cell packet.

33. The configurable network interface device of claim 29, wherein the set of statistics engines includes:

5 a cell statistics engine coupled to receive cell packets and maintain a record of payload types, cell loss priority, generic flow control, and header error control errors.

34. The configurable network interface device of claim 29, wherein the set of statistics engines includes:

5 an AAL statistics engine for a predetermined AAL type coupled to receive cell packets and maintain records of total number of cell packets containing a predetermined bit pattern and total number of cell packets containing the predetermined bit pattern that do not conform to the predetermined AAL type.

35. A method for processing a network packet from a communications network, said method comprising the steps of:

5 (a) transforming the network packet into a cell packet, wherein the cell packet includes a portion of the network packet and data derived from the network packet; and

- 64 -

(b) generating a set of statistics related to the cell packet.

36. The method of claim 35, wherein said step (a) includes the step of:

hashing a set of information in the network packet to obtain a VCC stream number.

37. The method of claim 36, wherein said step (a) includes the step of:

loading a set of data into the cell packet, based on the VCC stream number.

38. The method of claim 35, wherein said step (a) includes the step of:

loading a time stamp into the cell packet.

39. The method of claim 35, wherein said step (b) includes the steps of:

identifying a characteristic of the cell packet; and
incrementing a counter for the characteristic.

40. The method of claim 39, wherein the characteristic is a characteristic selected from a group consisting of generic flow control type, payload type, cell loss priority, and header control error.

41. The method of claim 35, wherein said step (b) includes the steps of:

determining whether the cell packet includes a predetermined bit pattern;

- 65 -

5 determining whether the cell packet is errored, if the cell packet contains the predetermined bit pattern; and

recording the result of said determination of whether the cell packet is errored, if the cell packet contains the predetermined bit pattern.

42. A computer readable medium having a set of instructions therein for enabling a computer to configure a network interface device to process a network packet from a communication network, said set of instructions comprising:

5 a first set of instructions, which when executed by the computer cause the network interface device to transform the network packet into a cell packet, wherein the cell packet includes a portion of the network packet and data derived from the network packet; and

10 a second set of instructions, which when executed by the computer cause the network interface device to generate a set of statistics related to the cell packet.

43. The computer readable medium of claim 42, wherein said first set of instructions includes:

5 a third set of instructions, which when executed by the computer cause the network interface device to hash a set of information in the network packet to obtain a VCC stream number.

44. The computer readable medium of claim 43, wherein said first set of instructions further includes:

5 a fourth set of instructions, which when executed by the computer cause the network interface device to load a set of data into the cell packet, based on the VCC stream number.

- 66 -

45. The computer readable medium of claim 42, wherein said first set of instructions includes:

a fifth set of instructions, which when executed by the computer cause the network interface device to load a time stamp into the cell packet.

46. The computer readable medium of claim 42, wherein said second set of instructions includes:

a sixth set of instructions, which when executed by the computer cause the network interface device to identify a characteristic of the cell packet; and

5 a seventh set of instructions, which when executed by the computer cause the network interface device to increment a counter for the characteristic.

47. The computer readable medium of claim 46, wherein the characteristic is a characteristic selected from a group consisting of a generic flow control type, pay load type, cell loss priority, and header error control error.

48. The computer readable medium of claim 42, wherein said second set of instructions includes:

5 a ninth set of instructions, which when executed by the computer cause the network interface device to determine whether the cell packet includes a predetermined bit pattern;

a tenth set of instructions, which when executed by the computer cause the network interface device to determine whether the cell packet is errored, if the cell packet contains the predetermined bit pattern; and

10 an eleventh set of instructions, which when executed by the computer, cause the network interface device to record the result of said determination of

- 67 -

whether the cell packet is errored, if the cell packet contains the predetermined bit pattern.

49. A configurable node for use in a network interface device, wherein the network interface device provides for coupling a network peripheral to a communication network, wherein said network interface device includes a system bus, said configurable node comprising:

5 a first processing engine for performing an operation using data in an input packet containing a set of information from the communication network, wherein said first processing engine includes a packet input for receiving the input packet and a packet output for outputting an output packet, wherein the output packet is derived from the input packet;

10 a data transfer controller for controlling the transfer of data between the configurable node and a data storage unit; and

a task controller coupled to the system bus of the network interface device, said data transfer controller, and said first processing engine, wherein said task controller controls communication between the system bus of the network interface device, said data transfer controller, and said first processing engine.

50. The configurable node of claim 49, wherein said configurable node is a field programmable gate array.

51. The configurable node of claim 49, further including:

a second processing engine coupled to said task controller, said second processing engine having a packet input coupled to the packet output of the first processing engine and a packet output for providing a packet derived from a packet received by said packet input of said second processing engine.

5

- 68 -

52. The configurable node of claim 51, wherein said task controller controls communication between the system bus of the network interface device, said data transfer controller, and said second processing engine.

53. A computer readable medium having a set of instructions stored therein for execution by a configurable node to cause the configurable node to be configured, wherein the configurable node is for use in a network interface device, wherein the network interface device provides for coupling a network peripheral to a communication network, wherein the network interface device
5 includes a system bus, said set of instructions including:

a first set of instructions, which when executed by the configurable node cause the configurable node to implement a first processing engine for performing an operation using data in an input packet containing a set of
10 information from the communication network, wherein said first processing engine includes a packet input for receiving the input packet and a packet output for outputting an output packet, wherein the output packet is derived from the input packet;

a second set of instructions, which when executed by the configurable
15 node cause the configurable node to implement a data transfer controller for controlling the transfer of data between the configurable node and a data storage unit; and

a third set of instructions, which when executed by the configurable node cause the configurable node to implement a task controller coupled to the system
20 bus of the network interface device, said data transfer controller, and said first processing engine, wherein said task controller controls communication between

- 69 -

the system bus of the network interface device, said data transfer controller, and said first processing engine.

54. The computer readable medium of claim 53, wherein said set of instructions further includes:

5 a fourth set of instructions, which when executed by the configurable node cause the configurable to implement a second processing engine coupled to said task controller, said second processing engine having a packet input coupled to the packet output of the first processing engine and a packet output for providing a packet derived from a packet received by said packet input of said second processing engine.

55. The computer readable medium of claim 54, wherein said task controller controls communication between the system bus of the network interface device, said data transfer controller, and said second processing engine.

56. A network peripheral for receiving information from a communication network, said network peripheral comprising:

a configurable network interface device for coupling the network peripheral to the communications network; and

5 a discovery engine coupled to said configurable network interface device for configuring said configurable network interface device, wherein said discovery engine includes:

means for configuring the network interface device for interpreting information,

10 means for determining whether the network interface device is configured for interpreting information received from the communication network, and

- 70 -

means for modifying the network peripheral based on the determination made by said means for determining.

57. The network peripheral of claim 56, wherein said configurable network interface device includes:

5 a packet creation engine for receiving a set of information derived from the communication network and transforming the set of information into a cell packet, wherein the cell packet contains the set of information and data derived from the set of information; and

a set of statistics engines coupled to receive the cell packet, wherein at least one statistics engine in said set of statistics engines generates a set of statistics related to the cell packet.

58. The network peripheral of claim 57, wherein said set of statistics engines includes:

5 a cell statistics engine coupled to receive cell packets and maintain a record of payload types, cell loss priority, generic flow control, and head error control errors.

59. The network peripheral of claim 56, wherein said set of statistics engines includes:

5 an AAL statistics engine for a predetermined AAL type coupled to receive cell packets and maintain records of total number of cell packets containing a predetermined bit pattern and total number of cell packets containing the predetermined bit pattern that do not conform to a predetermined AAL type.

60. The network peripheral of claim 56, wherein said means for determining include:

- 71 -

means for querying the configurable network interface device to obtain data; and

- 5 means for determining whether the configurable network interface device is configured for interpreting information received from the communication network, based on the data obtained from said means for querying.

61. The network peripheral of claim 56, wherein said means for modifying includes:

- 5 means for configuring the configurable network interface device to interpret information from the communication network in accordance with a predetermined protocol, when said means for determining indicates that the configurable network interface device is not configured for interpreting information from the communication network in accordance with a protocol employed on the communication network.

62. The network peripheral of claim 56, wherein said means for configuring includes:

- 5 a resource library coupled to said discovery engine, wherein said resource library is readable by said discovery engine and said resource library contains a set of instructions for execution by said discovery engine for configuring a node on said configurable network interface device, said set of instructions including:
- 10 a first set of instructions, which when executed by the discovery engine cause the discovery engine to implement a first processing engine in the node for performing an operation using data in an input packet containing a set of information from the communication network, wherein said first processing engine includes a packet input for receiving the input packet and a packet output for outputting an output packet, wherein the output packet is derived from the input packet;

- 72 -

15 a second set of instructions, which when executed by the discovery engine cause the discovery engine to implement a data transfer controller in the configurable node for controlling the transfer of data between the node and a data storage unit; and

20 a third set of instructions, which when executed by the discovery engine cause the discovery engine to implement a task controller in the node, wherein the task controller is coupled to a system bus of the configurable network interface device, the data transfer controller, and the first processing engine, wherein the task controller controls communication between the system bus of the configurable network interface device, the data transfer controller, and the first processing engine.

63. The network peripheral of claim 56, wherein said configurable network interface device includes:

a system bus; and

a first configurable node including:

5 a first processing engine for performing an operation using data in an input packet containing a set of information from the communication network, wherein said first processing engine includes a packet input for receiving the input packet and a packet output for outputting an output packet, wherein the output packet is derived from the input packet,

10 a first data transfer controller for controlling the transfer of data between the first configurable node and a first data storage unit, and

15 a first task controller coupled to the system bus of the network interface device, said first data transfer controller, and said first processing engine, wherein said first task controller controls communication between the system bus of the network interface device, said first data transfer controller, and said first processing engine.

- 73 -

64. The network peripheral of claim 63, further including a second configurable node, wherein said second configurable node includes:

5 a second processing engine having a packet input coupled to the packet output of the first processing engine and a packet output for providing a packet derived from a packet received by said packet input of said second processing engine;

a second data transfer controller for controlling the transfer of data between the configurable node and a second data storage unit; and

10 a second task controller coupled to the system bus of the network interface device, said second data transfer controller, and said second processing engine, wherein said second task controller controls communication between the system bus of the network interface device, said second data transfer controller, and said second processing engine.

65. The network peripheral of claim 63, wherein said first processing engine is a packet creation engine for transforming the packet received on the first packet input into a cell packet containing the a set of information from the packet and data derived from the set of information, wherein the cell packet is
5 provided on said packet output.

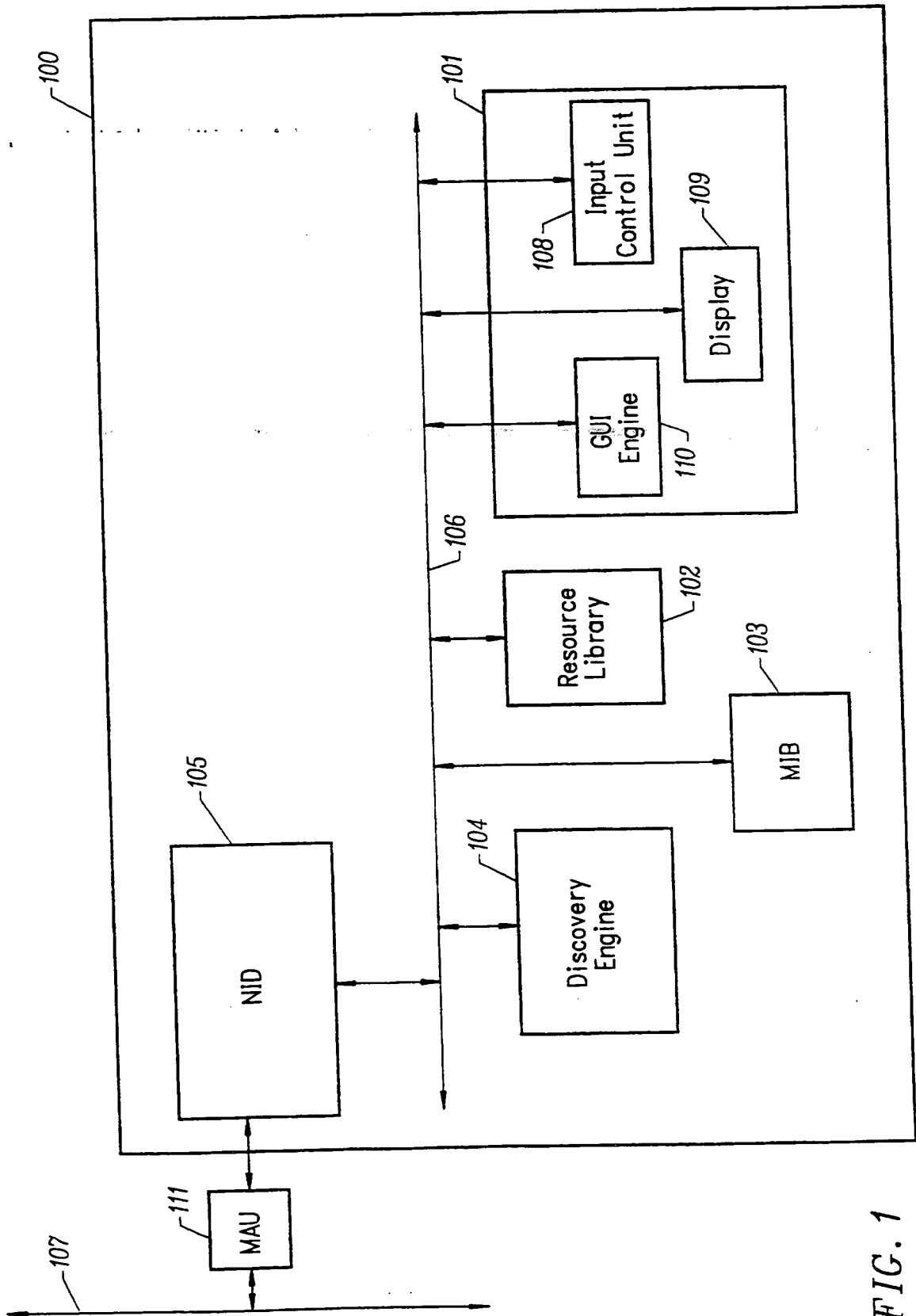


FIG. 1

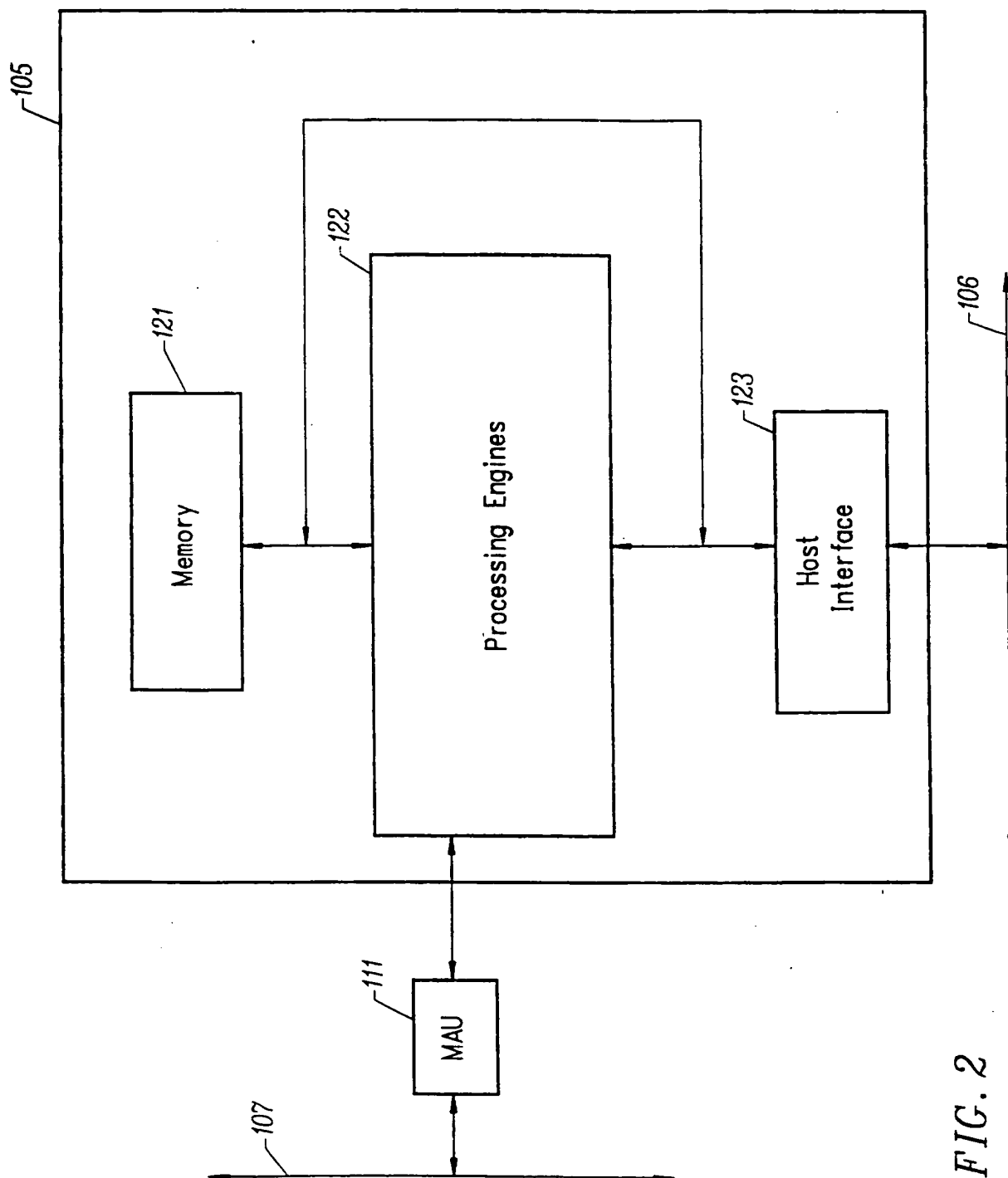


FIG. 2

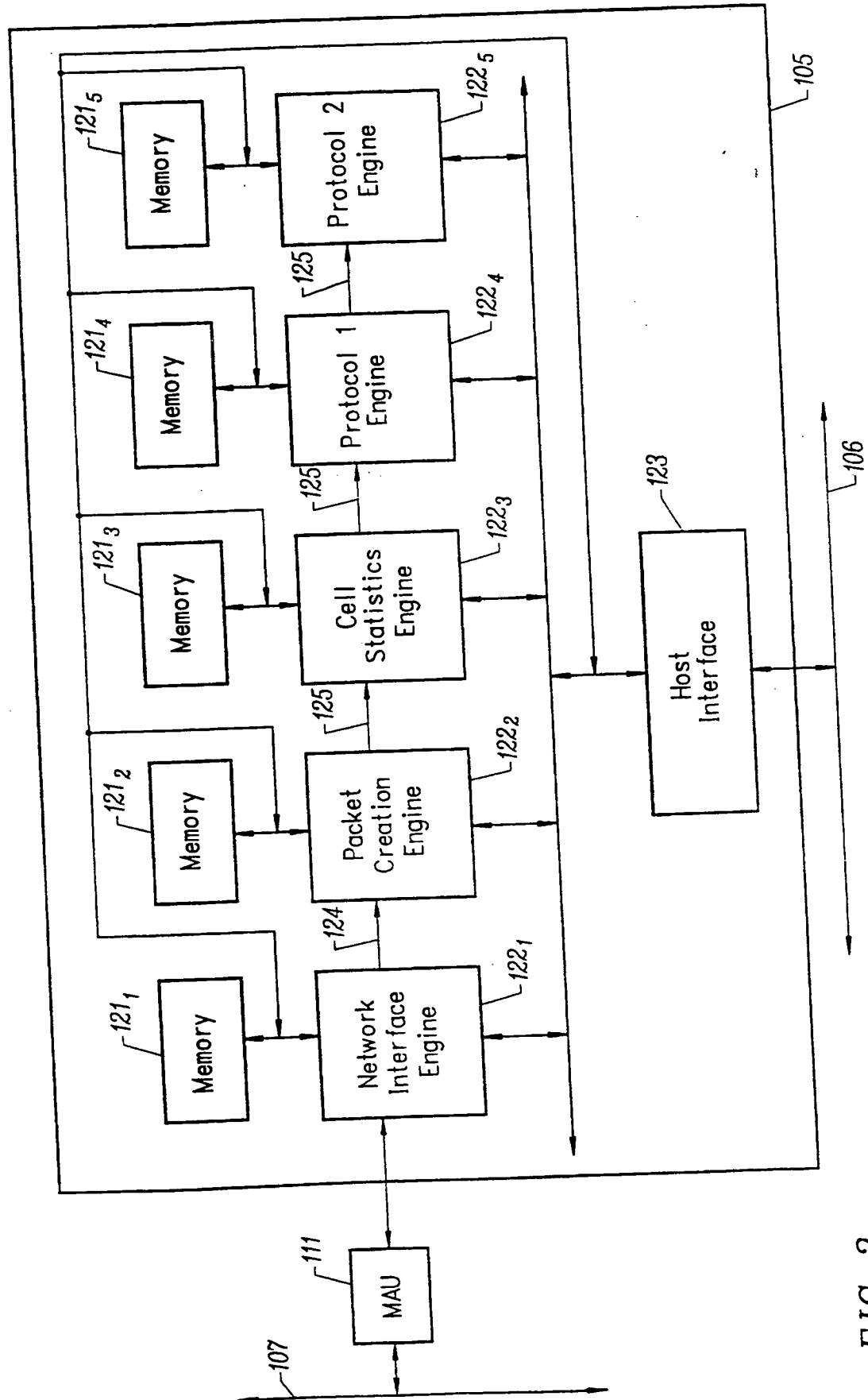


FIG. 3

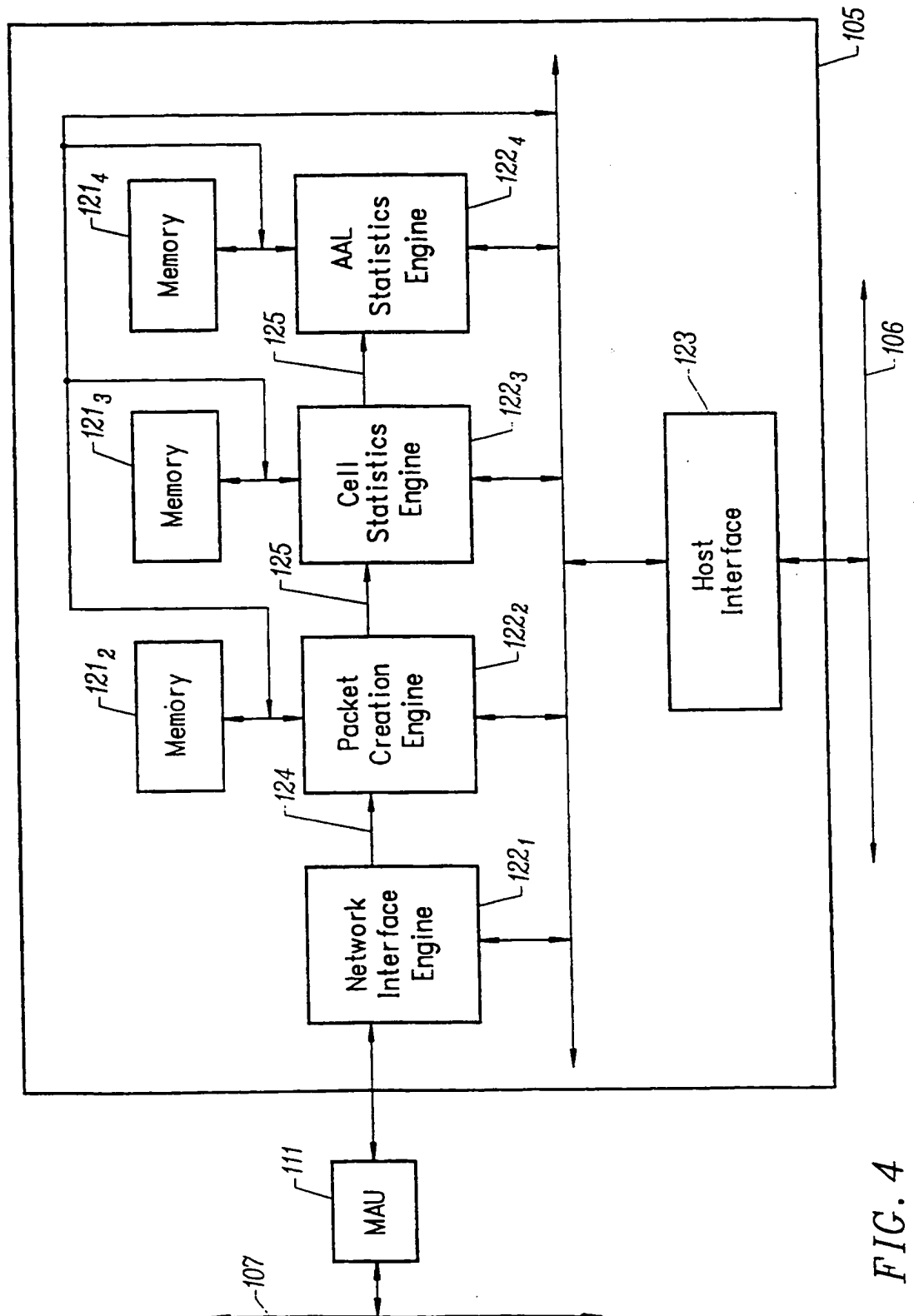


FIG. 4

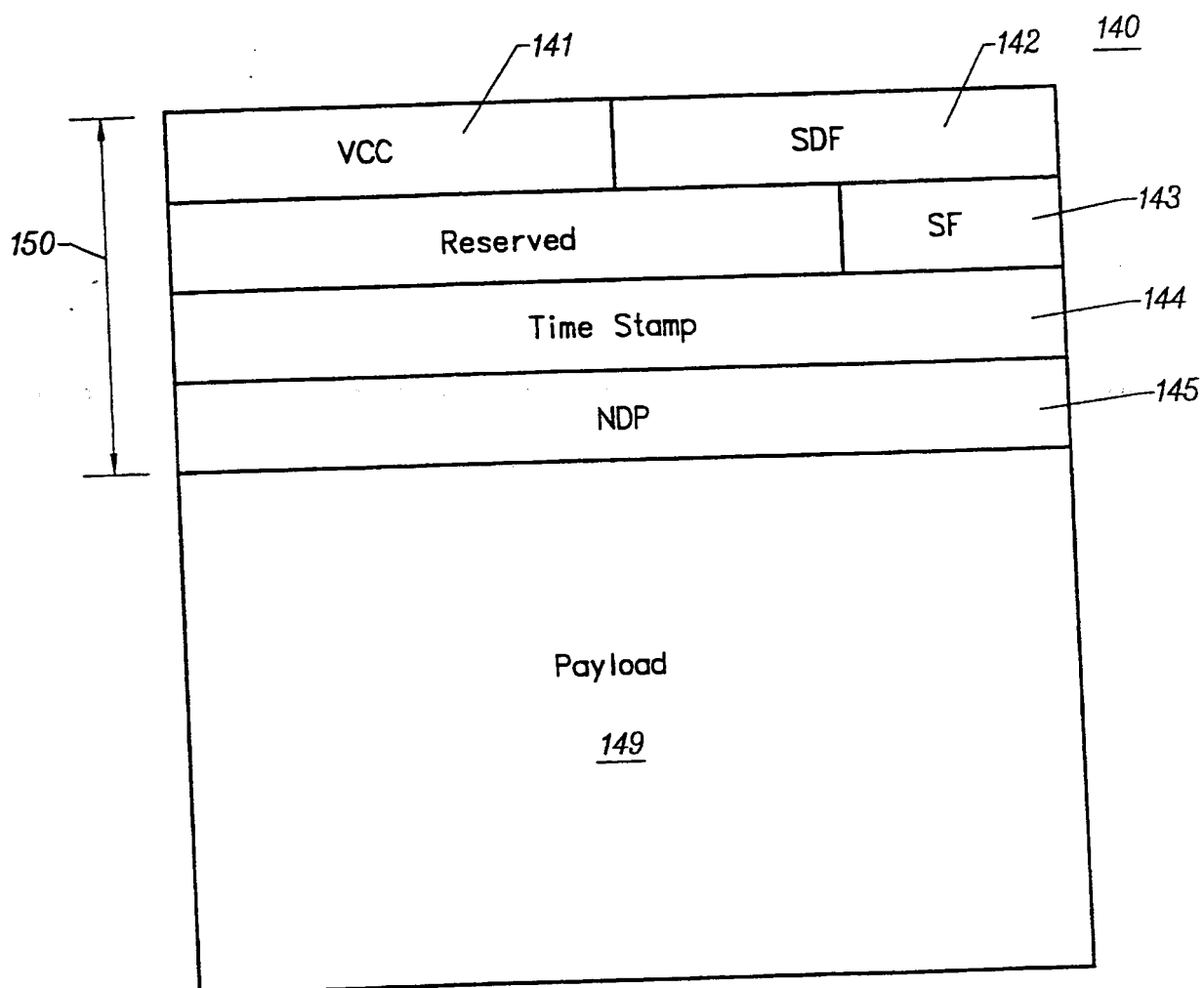


FIG. 5

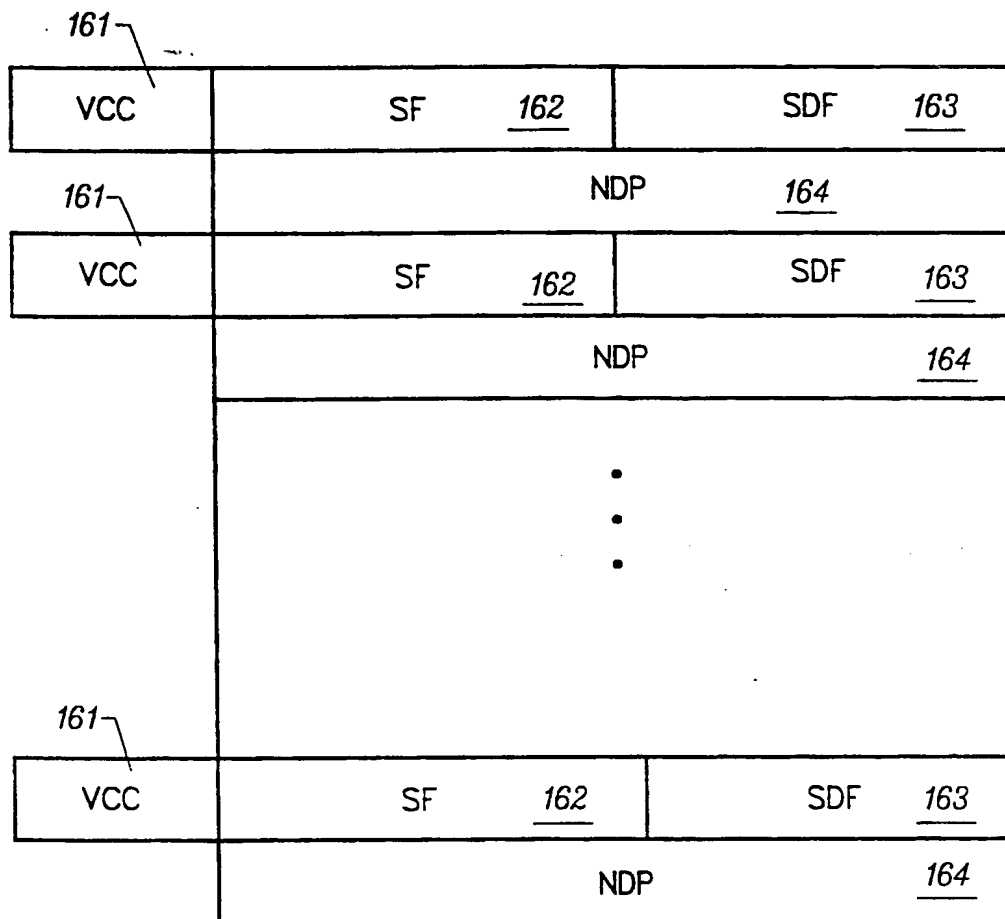
160

FIG. 6

165

<u>A</u>	<u>B</u>
166 ↗	↖ 167
GFC0	GFC0
GFC1	GFC1
GFC2	GFC2
GFC3	GFC3
GFC4	GFC4
GFC5	GFC5
GFC6	GFC6
GFC7	GFC7
GFC8	GFC8
GFC9	GFC9
GFC10	GFC10
GFC11	GFC11
GFC12	GFC12
GFC13	GFC13
GFC14	GFC14
GFC15	GFC15
Correctable HEC Errors	Correctable HEC Errors
Uncorrectable HEC Errors	Uncorrectable HEC Errors

FIG. 7

8/16

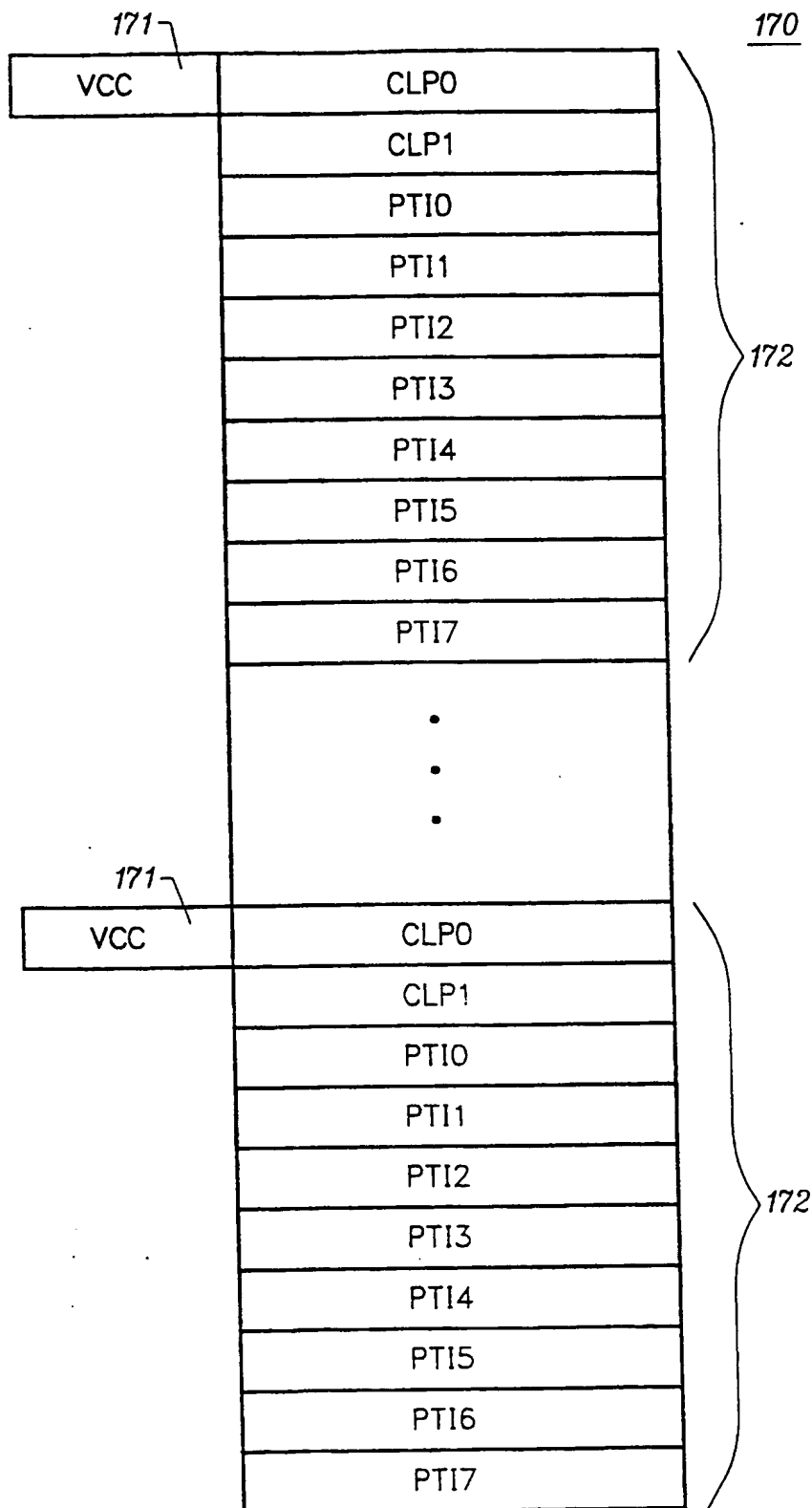


FIG. 8

9/16

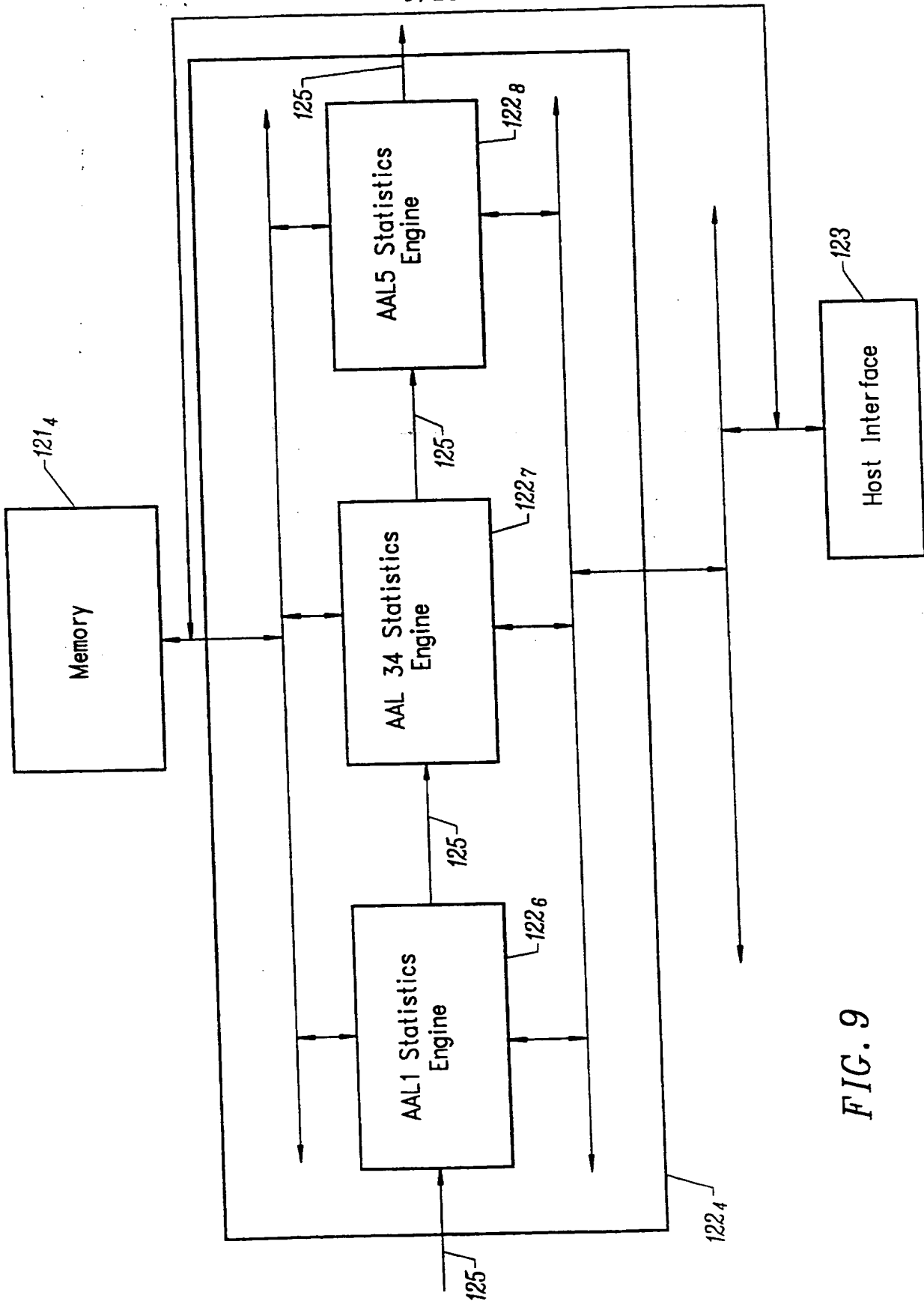


FIG. 9

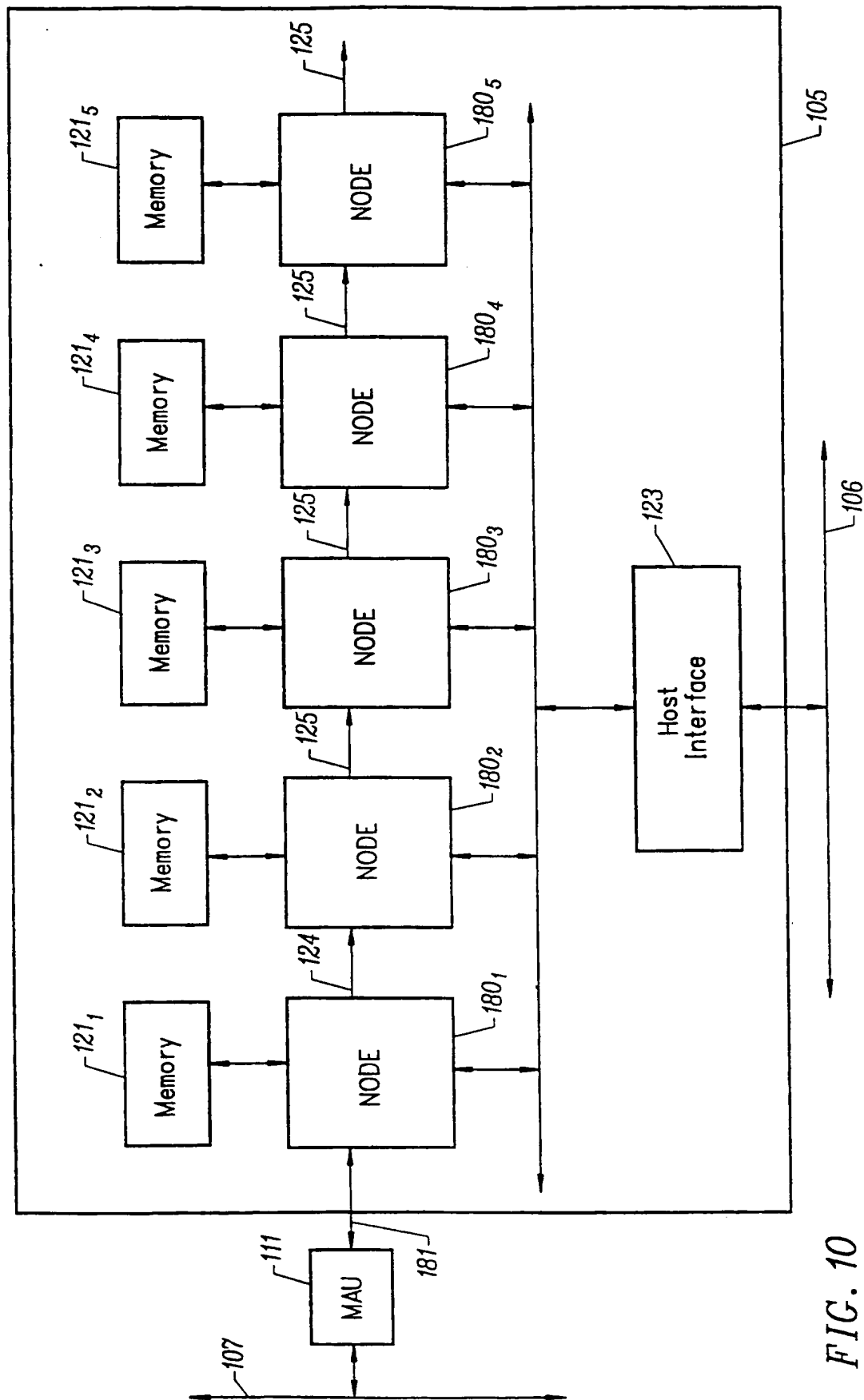


FIG. 10

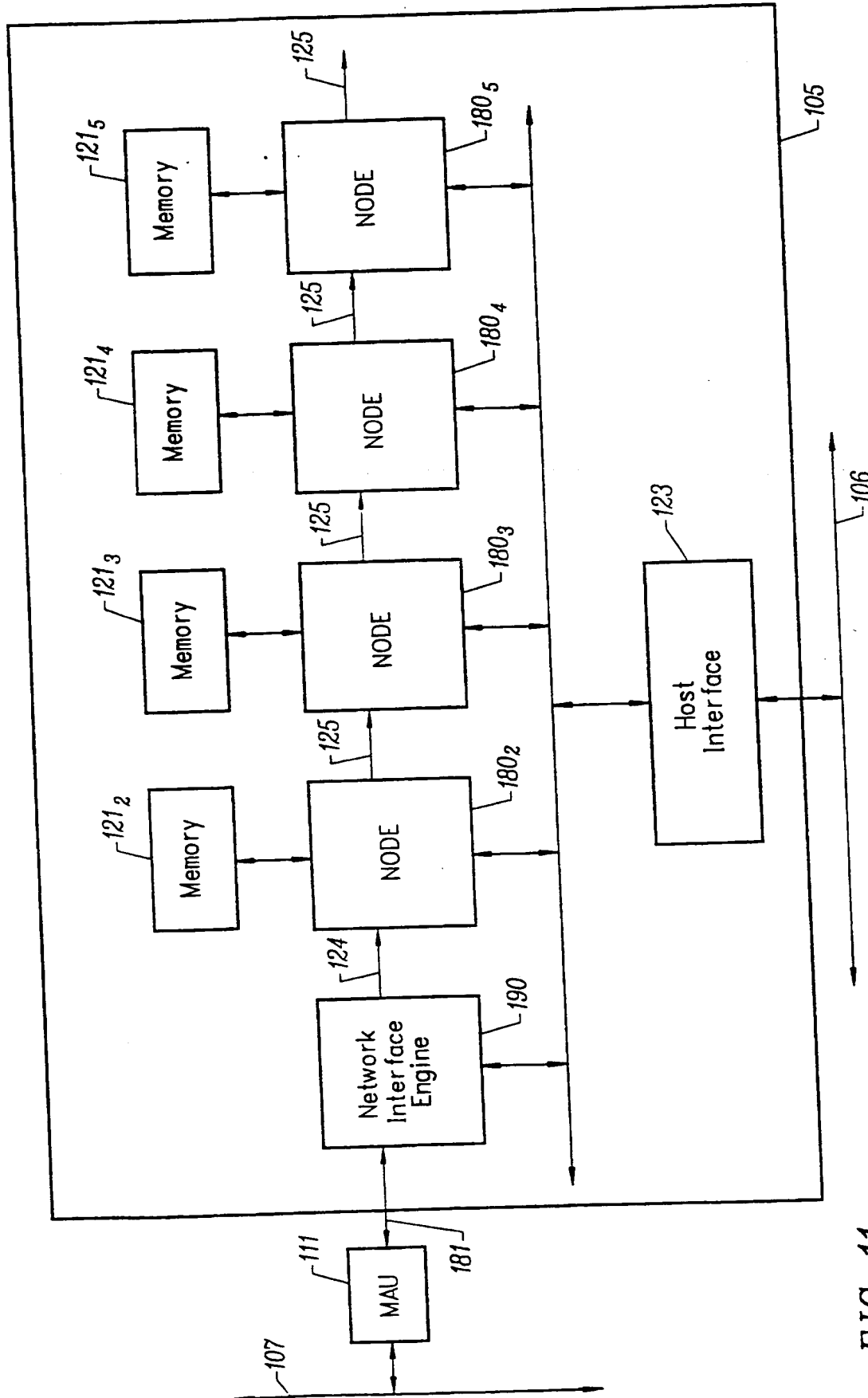


FIG. 11

12/16

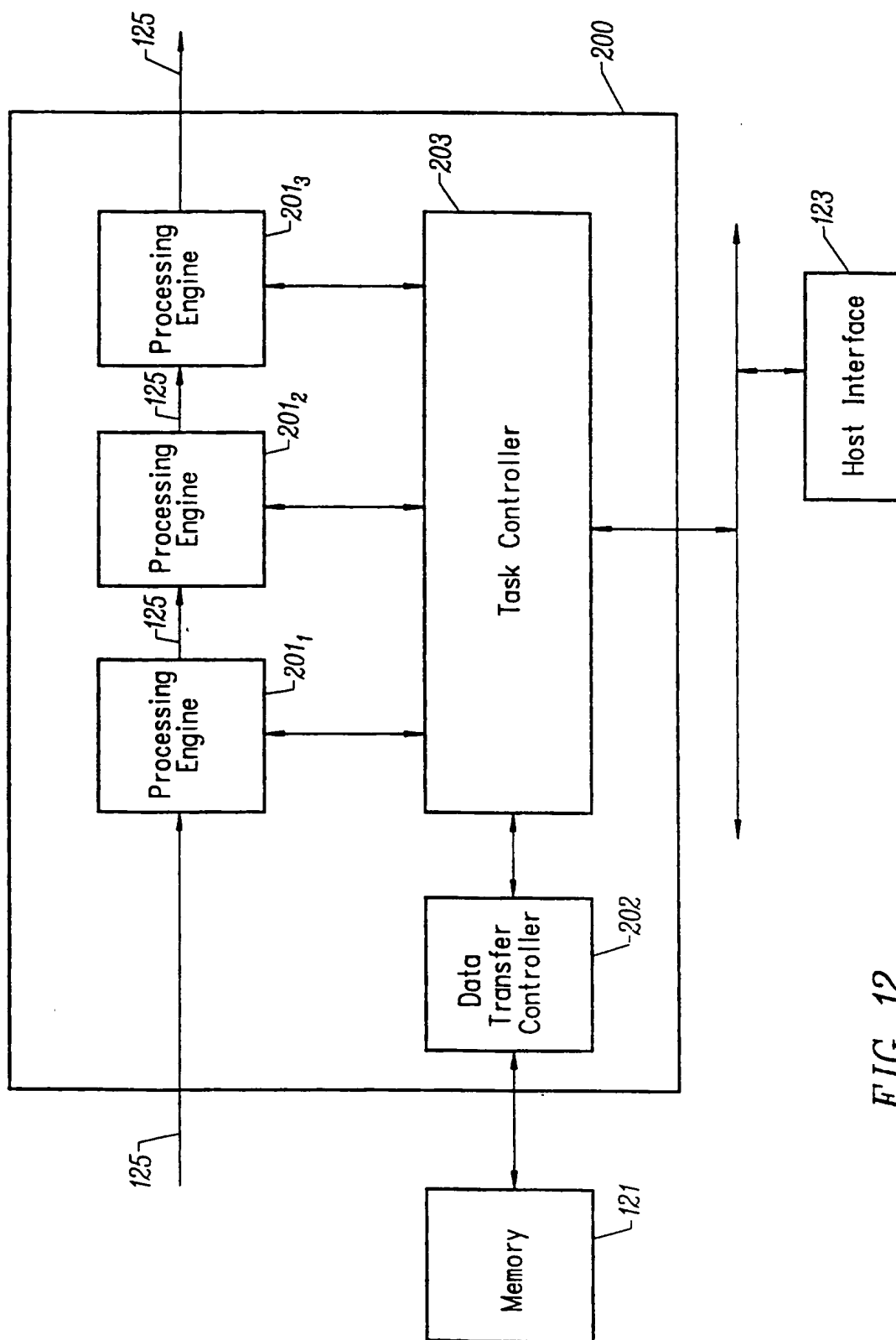


FIG. 12

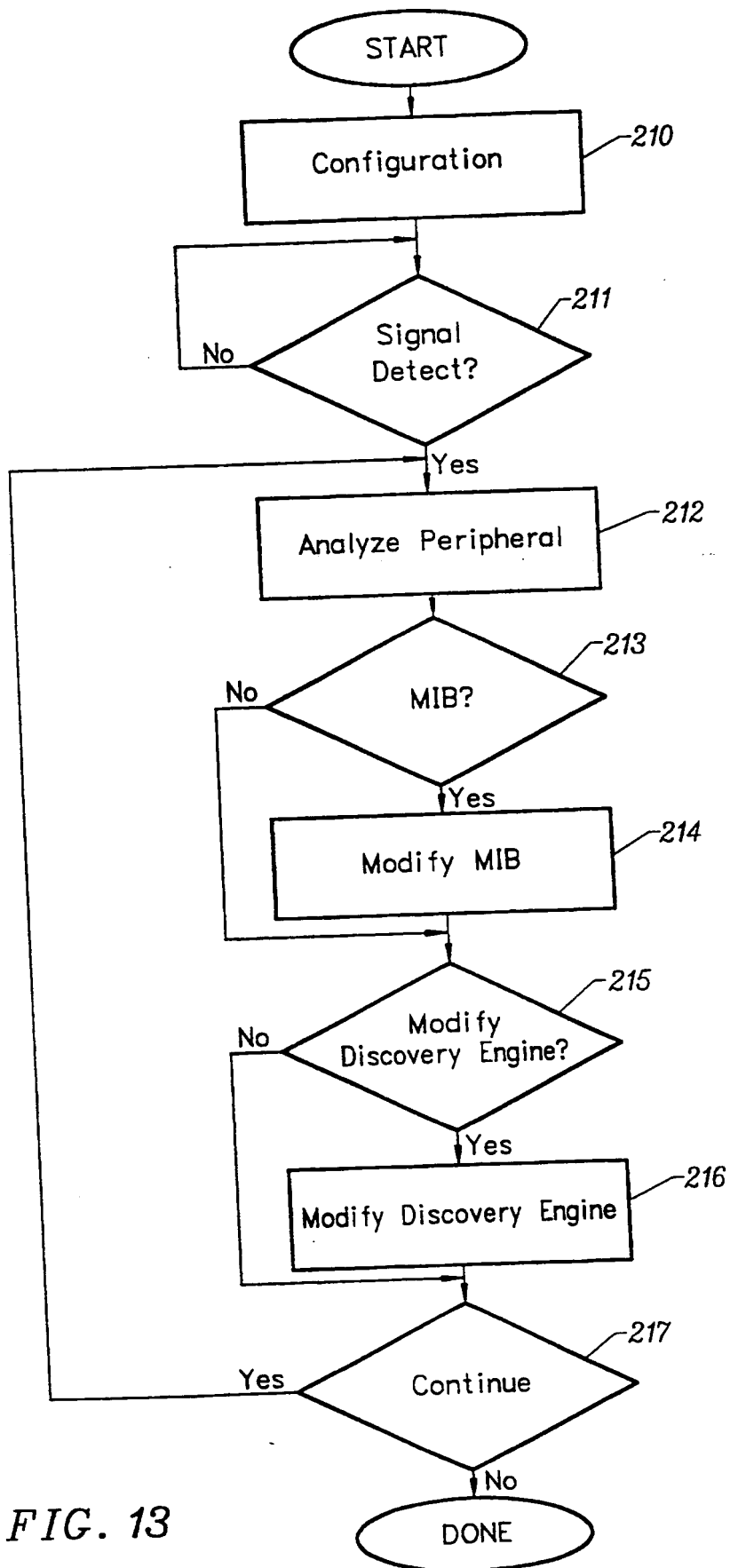


FIG. 13

14/16

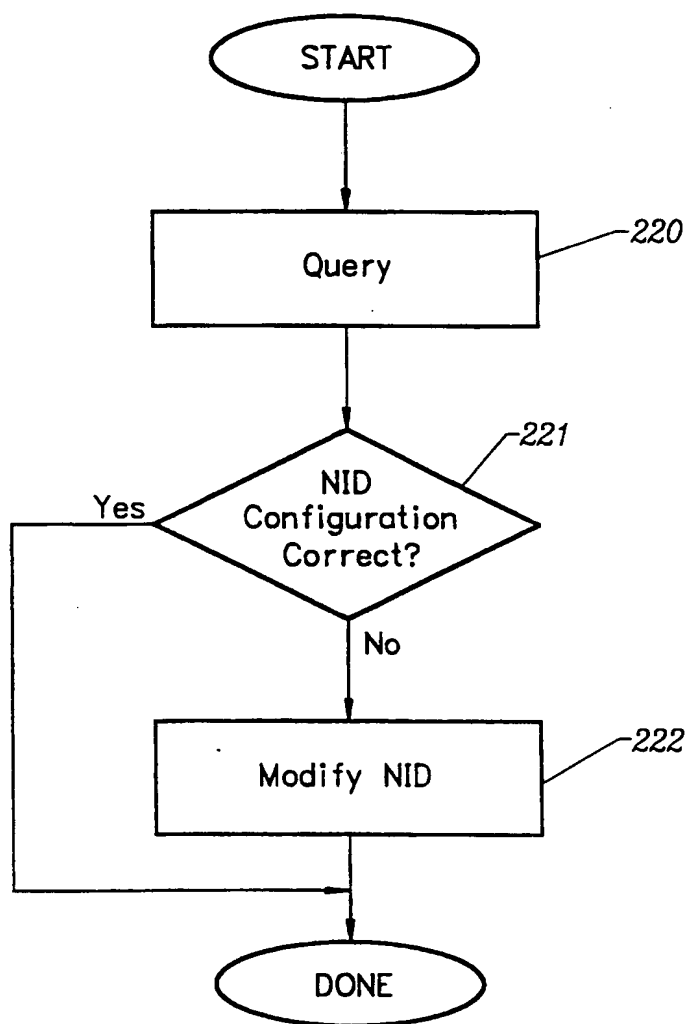


FIG. 14

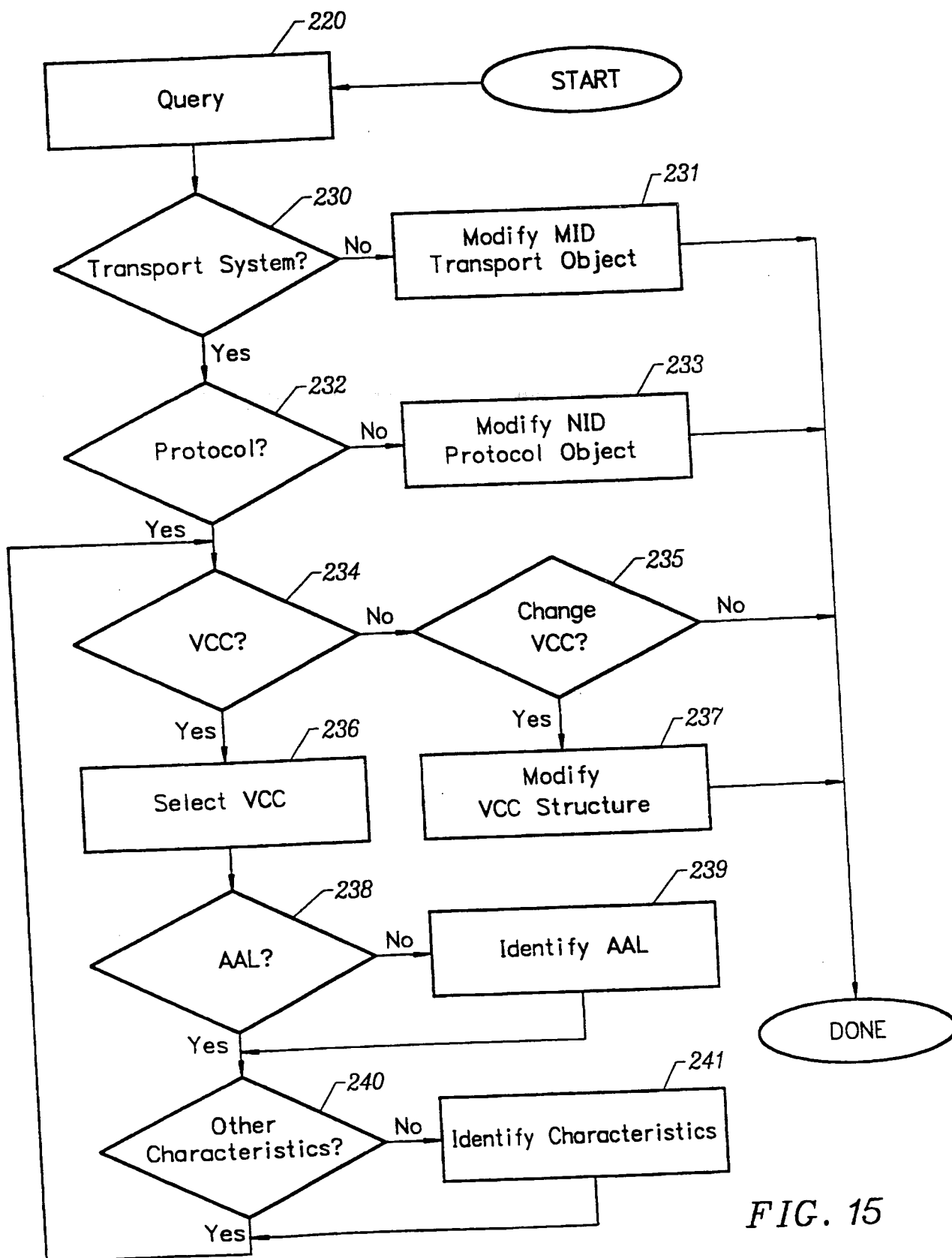


FIG. 15

16/16

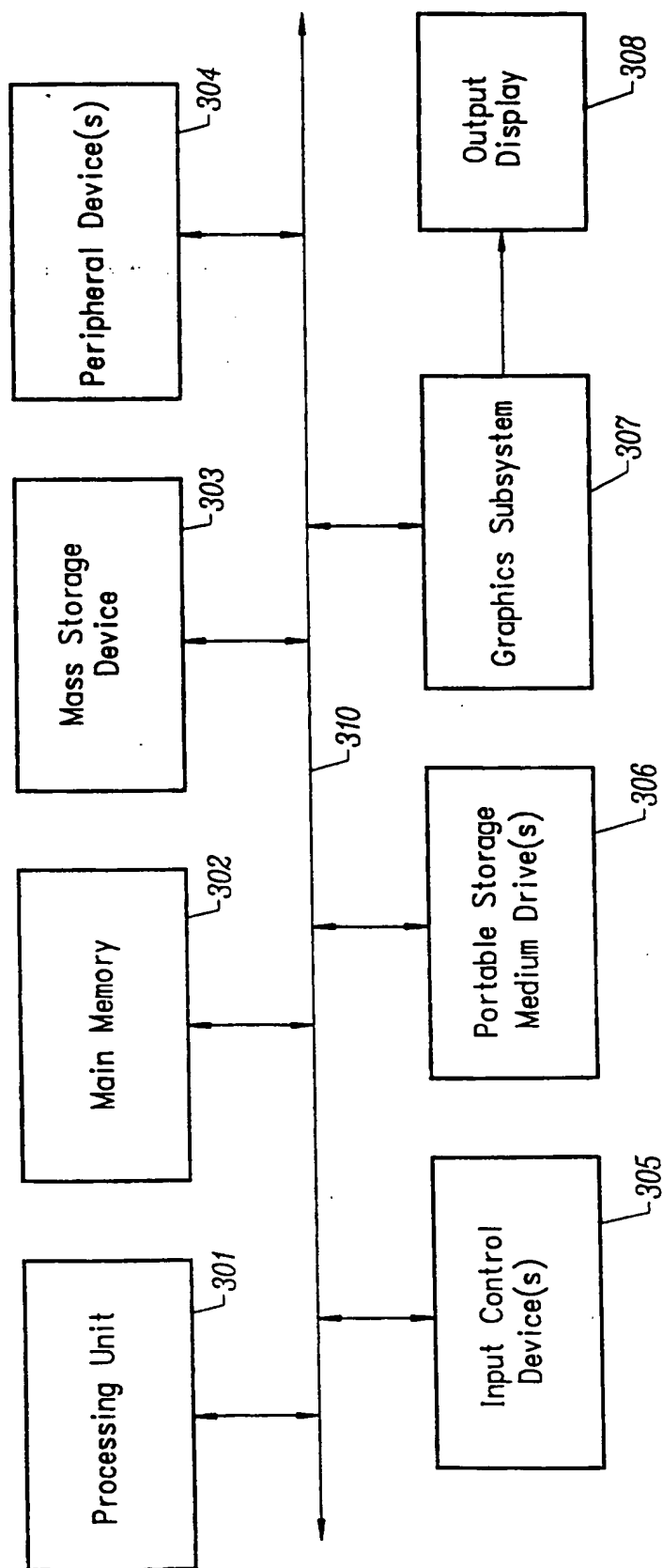


FIG. 16

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/11614

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) : G06F 13/00; H04L 12/56
US CL : 709/221, 224, 220; 710/8, 11, 15, 30, 65
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 709/221, 224, 220; 710/8, 11, 15, 30, 65; 370/395, 397

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
APS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,557,748 A [NORRIS] 17 September 1996, col. 1, lines 6-8, col. 2, lines 18-26, col. 4, lines 27-40, col. 5, lines 27-37, col. 8, lines 52-53, col. 11, lines 7-18.	1, 2, 7-9, 23-25
X	US 5,758,070 A [LAWRENCE] 26 May 1998, col. 1, lines 6-8, col. 3, lines 57-61, col. 11, lines 21-25, col. 17, lines 46-64.	1-3, 5, 23, 24
X	US 5,671,355 A [COLLINS] 23 September 1997, col. 2, lines 58-67, col. 3, lines 6-11, col. 4, lines 34-39, col. 9, lines 48-67, col. 10, lines 1-37, col. 11, lines 16-40.	1-3, 5, 10-15, 17, 23, 24, 26, 27, 56, 60, 61.
Y		4, 6, 13, 16, 18-22, 28, 57, 58

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:	*T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
A document defining the general state of the art which is not considered to be of particular relevance	*X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
B earlier document published on or after the international filing date	*Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	*G* document member of the same patent family
O document referring to an oral disclosure, use, exhibition or other means	
P document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

23 SEPTEMBER 1999

Date of mailing of the international search report

19 OCT 1999

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231
Facsimile No. (703) 305-3230

Authorized officer

THOMAS C. LEE

Telephone No. (703) 305-9717

Joni Hill

INTERNATIONAL SEARCH REPORT

 International application No.
 PCT/US99/11614

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,708,659 A [ROSTOKER et al] 13 January 1998, col. 9, lines 31-67, col. 18, lines 38-42, col. 43, lines 39-57	35-37, 42-44
X	US 5,751,698 A [CUSHMAN et al] 12 May 1998, col. 6, lines 40-50, col. 8, lines 35-46.	35, 42
--		-----
Y	col. 11, lines 1-45, col. 12, lines 22-51.	18-22, 28, 39-41, 46-48
Y,P	US 5,805,571 A [ZWAN et al] 08 September 1998, abstract	4, 6
Y,P	US 5,842,039 A [HANAWAY et al] 24 November 1998, abstract, col. 7, lines 38-49, col. 8, lines 1-12	13, 16
X	US 5,619,499 A [NAKABAYASHI] 08 April 1997, col. 2, lines 55-62, col. 8, lines 40-58, col. 10, lines 4-22.	29, 33, 35, 45
--		-----
Y		30-32, 38, 42, 57, 58
Y	US 5,726,985 A [DANIEL et al] 10 March 1998, fig. 6, col. 2, lines 1-9, col. 6, lines 6-13, col. 13, lines 31-37, col. 20, lines 1-10	30-32, 38, 45
Y,P	US 5,761,191 A [VANDERVORT et al] 02 June 1998, col. 13, lines 40-64, col. 15, lines 24-33, col. 16, lines 1-40	39-41, 46-48

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/11614

Box I Observations where certain claims were found unsearchable (Continuation of Item 1 of first sheet)

This international report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of Item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

Group I, claim(s) 1-48 AND 56-65, drawn to configuring peripheral.
Group II, claim(s) 49-55, drawn to processing and transferring peripheral input/output data.

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:
1-48 and 56-65

Remark on Protest

☐
☐

The additional search fees were accompanied by the applicant's protest.
No protest accompanied the payment of additional search fees.